

Hooked on the Amiga with Fred Fish

Amazing COMPUTING™

Your Original AMIGA™ Monthly Resource

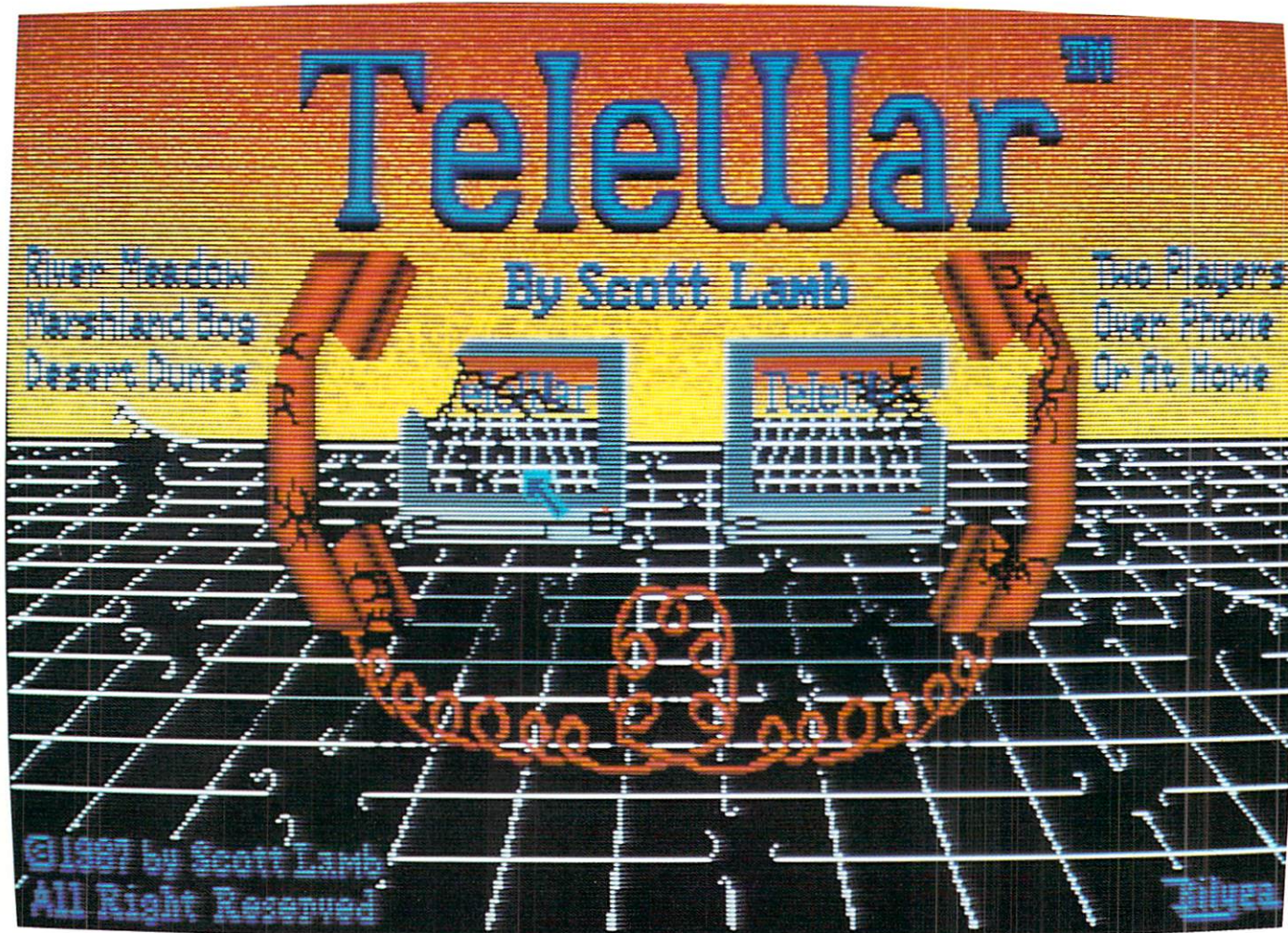
Volume 3 Number 2
US \$3.50 Canada \$4.50

LASERS!

AND
THE

AMIGA





THE ULTIMATE STRATEGY WARGAME

TeleWar allows you to use your computer and modem to play the ultimate strategy wargame over the phone, or on one computer in your home for \$39.95.



RIVER MEADOW

MARSHLAND BOG

DESERT DUNE

- Superb graphic war simulations
- 12 war scenarios
- Animated artillery, armor, and supply units with air support
- Smooth scrolling graphics
- 3D perspective terrain maps
- All moves graphically enacted on tele-connected computer
- Digitized sound effects
- Easy to use menus and requesters

Make Telecomputing A Blast You'll Love TeleWar!

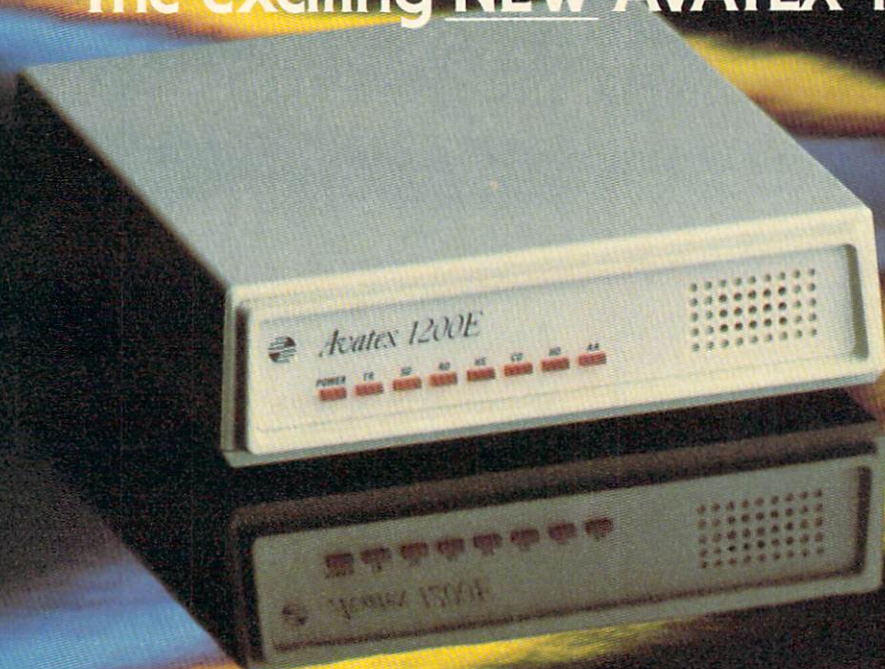
Published by Software Terminal
3014 Alta Mere, Fort Worth, TX 76116
817-244-4150 Modem: 817-244-4151

Available At Your Local Dealer

- Compatible with any modem
- 300, 1200, 2400, 9600 band
- Call originate or answer
- Null modem connect option
- Save game and transmit game options
- Opponent File Directories
- Send and receive typed messages with your human opponent
- Fully copyable to hard disk
- Upgrades available on BBS

GOOD THINGS COME IN SMALLER PACKAGES.

The exciting NEW AVATEX 1200 E.



\$85
TOTAL DELIVERED
PRICE.

AVATEX
Authorized Sales &
Service

ACTUAL SIZE ONLY 5" X 6"

FREE! Communication Software & CompuServe Access Time with each MODEM.

We give you a
great price ...

- + TOLL FREE ASSISTANCE
- + IMMEDIATE DELIVERY
- + NO CREDIT CARD SURCHARGE
- + NO SERVICE OR HANDLING FEES



FREE SHIPPING!

TOTAL HAYES COMPATIBILITY
8 LED STATUS INDICATORS
AUTO DIAL AND ANSWER
TONE OR PULSE DIALING

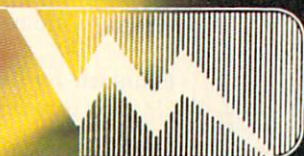
NEW 1200 E FEATURES

CCITT COMPATIBILITY
CALL PROGRESS DETECTION
INTERNAL SPEAKER
2 YEAR WARRANTY

Thousands of customers have purchased the AVATEX 1200 from Megatronics. New technologies now make it possible to offer an improved version of this popular Modem.

The 1200 E is amazing.
It weighs less and is smaller.
It has even more features.
It's still the same low price.

MEGATRONICS



MEGATRONICS, INC. P.O. BOX 3660, LOGAN, UT 84321

To Order

CALL FOR FREE CATALOGUE
CREDIT CARDS VERIFIED FOR YOUR PROTECTION

800-232-6342

INSIDE UTAH (801) 752-2642 FAX (801) 752-8752

We'll beat any advertised price.

MEGATRONICS, INC., P.O. BOX 3660, LOGAN, UTAH 84321

Amazing Features

- Laser Light Shows with the Amiga**
by Patrick Murphy 8
Lasers and the Amiga: A Dazzling Tandem.
- The Ultimate Video Accessory: Part III**
by Larry White 16
Take the final steps toward designing your own videos.
- Our First Desktop Video** by Larry White 19
A step-by-step guide to
organizing and presenting your first Amiga video.
- Hooked on the Amiga with Fred Fish**
by Ed Bercovitz 31
Inside views from the man behind all those "Fish" disks.
- Photo Quality Reproduction with the Amiga and
Digi-View** by Stephen Lebars 39
Stunning Digi-View images look great in hardcopy, too!
- Balancing Your Checkbook with WordPerfect Macros**
by Steve Hull 42
Hand your checkbook worries over to the Amiga.
- More Basic Text** by Bryan Catley 60
Displaying text on the screen can be even easier!
- Life: Part III** by Gerald Hull 81
The series winds up with the famed
nine-blit calculation and source to LIFER.
- Solutions to Linear Algebra through
Matrix Computations** by Robert Ellis 90
Simplify matrix algebra with basic operations & routines.

Amazing Departments

- From the Editor 4
Amazing Mail 6
Public Domain Software Catalog 105
Index of Advertisers 112

Amazing Columns

- The Amicus Network** by John Foust 72
A scrapbook look at the faces behind the Amiga.
- Roomers** by The Bandito 78
Amiga 3000, Virus news, and
"Will the real Laser Toaster please stand up?"
- Bug Bytes** by John Steiner 88
These bugs don't die off in the winter!
- Modula-2 Programming**
by Steve Faiwischewski 101
Catching up with Calc—a source follow-up.
- 68000 Assembler Language Programming**
by Chris Martin 104
Graphics!—Part II of Assemgr.asm.

Amazing Reviews

- Arazok's Tomb** by Kenneth E. Schaefer 23
"A terrifying adventure into the world of the occult."
- AiRT** by Steve Faiwischewski 27
An innovative icon-based programming language.
- Forms in Flight** by Steve Pietrowicz 51
Render and animate objects in 3D!
- Silicon Dreams and the Jewel of Darkness**
by Kenneth E. Schaefer 53
Hang on every word in these classic text adventures.
- Leisure Suit Larry** by Kenneth E. Schaefer 58
The ultimate nerd spends one racy night in the fast lane.
- Two New Entries From Microbotics** by John Foust 65
M501 Expansion & Starboard II MultiFunction board.
- Mindlight 7 and People Meter** by John Foust 68
Meet two unique Amiga products.
- Phantasie** by Kenneth E. Schaefer 74
Up your chances in this "monster mash" with the
Amazing Phantasie Character Editor.

35mm SLIDES FROM YOUR ARTWORK!



Professional 35mm Slides

- ◆ Now you can have reproduction and presentation quality slides of your work
- ◆ Distortion-free—fills in raster lines
- ◆ crisp bright colors, converts all IFF files

Now Custom graphic art and illustration.

\$10 each for your 1st to 4th slides.
5 to 9 slides—\$8.50
Over 10 slides—\$8.00
Add \$2.00 for shipping.
New York residents add sales tax.

Call (212) 777-7609 FOR DETAILS

Ask for Irene—or write TRU-IMAGE
P.O. Box 660, Cooper Station
New York, N.Y. 10276

Amiga is a trademark of Commodore-Amiga.

Amazing COMPUTING

Publisher:	Joyce Hicks
Circulation Manager:	Doris Gamble
Asst. to the Publisher:	Robert James Hicks
Corporate Trainer:	Virginia Terry Hicks
Traffic Manager:	Robert Gamble
Managing Editor:	Don Hicks
Submissions Editor:	Ernest P. Viveiros Jr.
Hardware Editor:	Ernest P. Viveiros Sr.
Music & Sound Editor:	Richard Rae
Art Director:	Keith Conforti
Illustrator:	Brian Fox
Advertising Manager:	John D. Fastino
Senior Copy Editor:	Michael T. Cabral
Copy Editors:	Karen P. Granger
	Julie Landry
Amicus & PDS Editor:	John Foust
Production Manager:	Mark Thibault
Production Assistants:	Rico A. Conforti
	Keven Desmarais

Advertising Sales & Editorial
1-617-678-4200

Special thanks to:
Buddy Terrell & Byrd Press
Betsy Piper at Tech Plus

Amazing Computing™ (ISSN 0886-9480)
is published by PIM Publications, Inc.,
P.O. Box 869, Fall River, MA 02722.
Subscriptions: in the U.S. 12 issues for
\$24.00; in Canada & Mexico, \$30.00;
Overseas: \$35.00. Printed in the U.S.A.
Copyright© 1988 by PIM Publications, Inc.
All rights reserved.

First Class or Air Mail rates available
upon request.

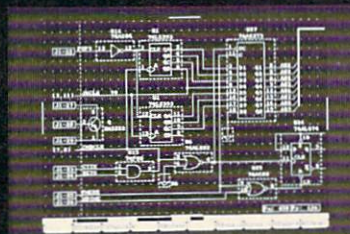
PIM Publications, Inc. maintains the right
to refuse any advertising.

PIM Publications, Inc. is not obligated to
return unsolicited materials. All materials
requesting return must be received with
a Self Addressed Stamped Mailer.

Prolific Inc.

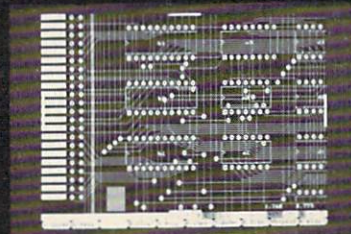
Available
Now!

PRO-NET



\$475.00

PRO-BOARD



\$475.00

For
AMIGA™
only

POWER+SIMPLICITY=PRODUCTIVITY

Feel our POWER...

PRO-NET:

- Variable template size
- A to E paper size
- Extensive Library included
- Auto device number with Zone control
- Gate swapping
- Single click change to negative logic
- Innovative weight assignment
- Auto page reference
- Dynamic error checking
- Creates BOM, Spare Part List, Net List, Error report, etc.
- Supports printers & plotters
- Supports Laser Printer
- Back annotation from PCB layout
- Move, Rubber... and lots more

See our SIMPLICITY...

Intelligent Function Keys make our programs extremely user friendly, provide maximum screen area, always display all relevant commands, avoids excessive cursor movement and screen flashing between menu & drawing, guides user through operation, minimizes training time.

PRO-BOARD:

- .025 inch grid
- .001 inch grid Library
- 12 mil trace, 13 mil space
- Produce 1, 2, 4 layer PCB
- Provides silk screen
- Auto coordinates assignment
- Auto produces power and grid layers
- Single line auto route
- Optional Net List Input for guided route, no need to look at schematic
- Prioritized route
- Dynamic error checking
- Supports printers, plotters and Gerber photo plotters
- Copy, Repeat... and lots more

TRY OUR DEMO DISKS FOR \$15 EACH.



Also from Prolific Inc.,

4 full feature AMIGA™ Macro Cross Assemblers
for Z80, 6809, 8085 & 8051.

See our SIMPLICITY

PRO-ASM

Feel our POWER

- Includes multi-pass Assembler, Linker, and Serial Down Load
- Generates relocatable Object Code Module
- Nested Macro
- Includes Files



\$85.00

Feel our POWER

- Conditional Assembling
- Rich set of directives
- Global and External Variables
- Data format includes Binary, Motorola Hex, Intel Hex, and Tek Hex

DEALER INQUIRY INVITED.

AMIGA™ trade mark of
Commodore Inc.

From The Editor:

Dear Amazing Computing

It is with some disappointment that I view the few programs published in your magazine. It is not given to all of us to have available C compilers, mainly due to the prohibitive costs of American software caused by the imbalance between the Australian and US dollars.

You could effect greater usage of these C programs if they were also listed in AmigaBASIC, and so give greater appeal to your magazine. My interest lies mainly in the area of Utility-type programs.

Since your publication is exclusively for the Amiga, may I suggest that you broaden the scope of the programs so all Amiga owners can use them, bearing in mind that these computers came equipped with AmigaBASIC?

A.N. Stephens
Australia

We agree!

Amazing Computing™ has been publishing AmigaBASIC articles since our second issue, V1.2 (March 1986). We would have published sooner, except only ABASIC was available for our premiere issue! AC does not publish every AmigaBASIC article received, since not every AmigaBASIC program/article demonstrates broad programming examples or unique approaches.

However, we are continuously searching for the program or article which can broaden the user's knowledge of the Amiga. If this outlet is AmigaBASIC, we are even more receptive, since every Amiga owner has access to this entry level programming language.

As for supplying both C and AmigaBASIC versions of a program, that is not always possible. All published programs are submissions by Amiga users with individual language preferences. A program is usually a personal project the programmer has spent a great many hours developing. It is unusual to find an author willing to commit the extra hours required to provide a multi-lingual article.

However, take heart; this issue has such an article. "Solutions to Linear Algebra through Matrix Computations" (page 90) is supplied in both AmigaBASIC and C.

Alternative Amiga languages are also important. Not only do we receive many requests for each user's personal favorite language, but we believe readers should be aware of the differences in programming environments. By being aware of the good and bad points of a language, as well as whatever options are available, users can pick the programming tool that best fits their styles and the demands of their projects.

"The Big Picture": Next month

Due to deadline pressures, our on-going column, "The Big Picture," was not in final form at press time. More time was needed to produce assembly code in legible form. Warren Ring's assembler series returns next month.

Our Second Anniversary

With a great deal of pride and a bit of amazement, we are publishing our twenty-third issue of *Amazing Computing™*.

We are proud that we have been able to produce an *Amazing Computing™* each month exactly as we planned. Our entire philosophy has been to produce a magazine we would want to read. It appears a great many other Amiga owners were interested in the same things we found interesting. For this, we are extremely thankful.

The time has passed so quickly! It seems only a short while ago our first Amiga arrived. A great deal of time, many overnight sessions, press changes, telephone calls, etc. went into creating the rather large *Amazing Computing™* back issue collection.

As always, thank you. We never forget the support of our readers and authors. *Amazing Computing™* is a combined effort from a great many dedicated individuals. Each issue is a compilation of the best and most timely articles we could produce, and a little more. We produce each article to be an asset today and a great value tomorrow. We have a saying at PiM Publications, Inc.: "We are not here to make a killing. We are here to make a living." Thank you.

Sincerely,

Don Hicks
Managing Editor

Amazing Dealers

The following are Amazing Dealers, dedicated to supporting the Commodore-Amiga™. They carry Amazing Computing™, your resource for information on the Amiga™.

If you are not an Amazing Dealer, but would like to become one, contact:

PiM Publications, Inc.
P.O.Box 869
Fall River, MA. 02722
1-617-678-4200

Alabama About Data Systems Alabama Book Store Command Computers Electronics Boutique Mail Tel Communications Main Photo & Computer Shop Universal Computer System	Huntsville Tuscaloosa Birmingham Hoover Mobile Montgomery Mobile	Personal Computer Center Software City Software Kingdom Spectrum Computers Trioam Computers <u>Dallas</u> Castle Video <u>Florida</u> A A Computers Commodore County Inc. Computer Bar Computer Base Computer Image Computer Specialty Computer Terminal Inc. Computers 4 Rent Computers Plus Cronics Inc. Data Base Education Computers Etc. Florida Book Store Heartland Computers Jill's News, Inc. K&L Computers Megaport Computer Center Micro Computer Center New Age Electronics Program Store Random Access Computers Inc. Software City Software Solutions Sunny's At Sunrise The Open Door The PC Collection <u>Georgia</u> Future System Software South Inc. The 64 Store <u>Hawaii</u> Aero Computers & Electronics Computer House Inc. J & C Repair <u>Idaho</u> A.B.I. Computer & Video Basis Computer & Video Inc. <u>Illinois</u> A Batcher ComputerLand of Illinois ComputerLand of Ohio DigitalWorld I.C.V. Data Center Inc. Irm Microsystems Inc. Krepps & Breen's Micro Age Software Or Systems Software Plus West The Memory Expansion Unique Computer <u>Indiana</u> Burkat Computer Cr. Byrnes of Angola Computer Corner Computer Corner Computer People Inc. Ging Chaney Indiana News Micro Computer Centers Inc. Spectra Computer Sys. Inc. Verna Computers <u>Iowa</u> Omnet Computers Micro Computer Applications LTD <u>Kansas</u> J & L Electronics McKinnis Computers Midwest Computers Software City <u>Kentucky</u> Computer Dimensions Inc. Computer Store of Kentucky Computer Outlet Reality World Computers <u>Louisiana</u> Computer Time Modern Business Machines Software Center International Software Solutions Software Source The Bok Sheriff The Computer Clinic <u>Maryland</u> Burned Treasures Computersware TODAY CompuVision Vision Center Layhill Newstead Micro Computer Center Software Advantage Software N Things Washburn	Norwich Orange East Windsor New Haven North Windham Newark Jacksonville Pineallas Park Pineallas Brandon Computer Image Melbourne Sebring W. Palm Beach St. Daytona Stewart Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louisville Lexington Shreveport Metairie Metairie Metairie Lafayette Sioux Lafayette Rockville Gettysburg Columbia Kensington Baltimore Rockville College Park	<u>Massachusetts</u> Club Computer D-Five DOX General Computer Store Hanscom Computer Center LCA Video & Computer Cr. Memory Location Omni-Tek Computers Pioneer Valley Data Equipment Softpro The Bit Bucket The Software Haus Tosam Inc. <u>Michigan</u> Computer Barn Hands On Computing Radio Shack Associate Stores Horse Tallahassee Sarasota Gainesville Mulberry Miami Edmunds Computer Center Venice Largo St. Petersburg Greensboro Ft. Walton Beach Software City Plantation Sunrise The PC Collection Cocoa Coral Gables Augusta Savannah Atlanta Axiis Horseshu Axiis Nampa Boise Rockford Park Ridge Niles Addison Mundelein Naperville Chicago Oakland Fairview Hts. Wheeling Hanover Park Bloomington Sterling South Bend Angola Fort Wayne Computer Corner Michigan City Greenfield Indianapolis Indianapolis Evansville West Lafayette Dex Maine LTD Marshalltown Liberal Nevada Marshalltown Overland Park Lexington Louisville Louis
---	--	--	--	---	---	---	---	---	---	---

Amazing Computing™ is also available in most B. Dalton Booksellers, B. Dalton Software Stores, Crown Books, and Software Etc. locations.

Amazing Mail:

Dear Amazing Computing

At last! A magazine which gives me the information I need about the Amiga. You deserve many congratulations for a well-produced, informative magazine.

One cry for help—I seem to be the only person in Europe who programs in FORTH. This cannot be true, can it? I leave messages on UK bulletin boards and haven't had a single reply. The FORTH tutorial in your magazine seems to have bitten the dust as well. Please tell Jon Bryan that someone reads his work and appreciates his efforts. You have a duty to release all those poor souls forced to program in 'C' and dream about nested parentheses into the wonderland of full machine control only easily available through FORTH.

I would be very interested in communicating with any FORTH users wherever they may be (Alpha Centaurii?) and would much appreciate your printing this letter.

Well done to all at PiM and a prosperous 1988.

Yours Forthly,
Dr. Richard Holder
32 Southridge Rise
Crowborough, East Sussex
TN6 1LJ

Thank you for the warm letter. Although I am at a loss concerning your comment of "at last." This is our twenty-third issue, as well as our second anniversary issue. We were the first monthly Amiga magazine, but we are still working hard to provide you with what you need to be a more informed Amiga user.

On the question of Mr. Bryan and Forth, Jon Bryan is a very active Amiga user, father, engineer, Compuserve Sysop, programmer, writer, and husband (not necessarily in that order). He has been providing insightful Forth programming as often as he can, but he is only one man!

We are open to any submissions which help expand our collective knowledge of Forth on the Amiga.

Jon has promised to keep producing as rapidly as he can, but any other interested parties who have developed Forth programs on the Amiga are invited to submit their articles to Amazing Computing™.

Dear Amazing Computing

Boy, what timing. I read your article on viruses and later that night SURPRISE, I find one on a disk of mine. To KILL the virus and to check your disks, can I make the following suggestions:

1. From PLink (others may have it, too) download two files. One is called VirusCheck by Bill Koester and the other is called VirusTest by Craig Bowen.

2. Make sure you have a VIRUS FREE Workbench (by using the two programs) and then copy both programs to your clean workbench startup-sequence. This way, every time you warm or cold boot, it will check both your system and the Workbench.

Another tip is to rename VirusCheck to V. That way you save a few keystrokes when you are checking all of your disks (what a pain!).

Eric Cook
Texas

Thank you! As an Amiga user, it is EXTREMELY disturbing to discover your Amiga is ALIVE! at 3 AM.

AMICUS Disk 24 contains several programs and utilities to find and secure the Amiga Virus (as outlined in John Foust's "AMICUS Network" column ACV3.1). This could be a great source of information to further the "cure."

Dear Amazing Computing

Why don't you have MasterCard? I took 1/2 hour at my bank and they charged a \$6.00 fee. Surely the additional percent MC charges would more than make up for the extra subscriptions you would receive. I doubt if I'll go through this again next year.

Peter Kinross
Australia

We would gladly have plastic! Unfortunately, MasterCard and Visa are not used by Amazing Computing™ because we have not been able to receive authorization. We have approached all the sources in our area, repeatedly been offered the service, and then turned down. Our "Yankee Bankers" are put off by our position as a subscription base, mail order establishment.

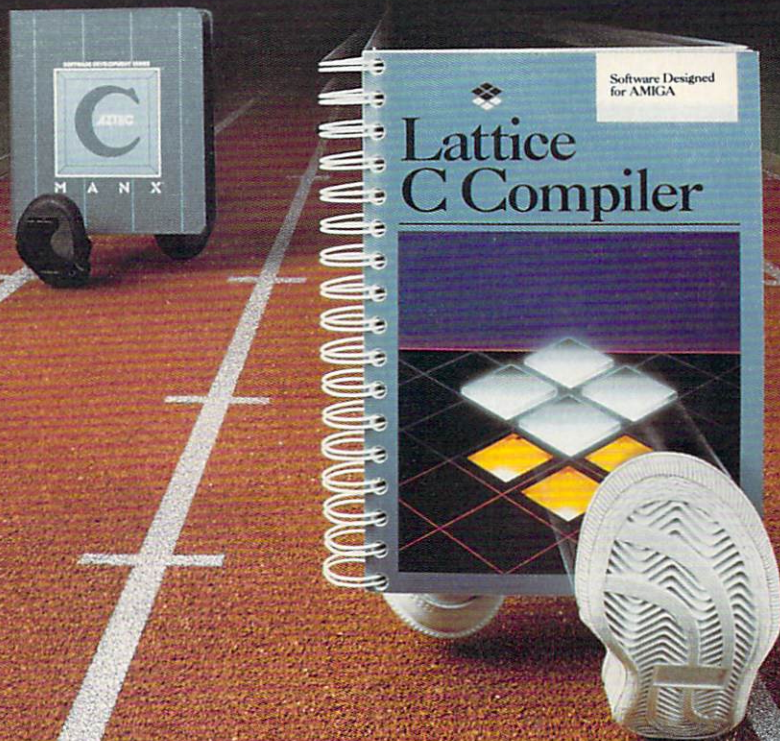
The most often heard complaint from our banking consultants is, "Our recent rules prohibit us from assigning any new customers who deal exclusively in mail order and/or subscription business." In a land filled with kitchen appliances, diet programs, and record sets all available only by mail order with a Visa or MasterCard, we wonder how these rules are administered.

Amazing Computing™ and PiM Publications have done some growing and a little long term planning. We now have an official "Office" and a new warehouse for shipping. These additional expenses were necessary for us to continue to grow with the Amiga. The added expense is more than warranted by our new ability to handle problems and shipments faster.

Since we are now in an office and warehouse environment, we may have a better chance at establishing a MasterCard and Visa service for our readers. (This seems to appeal to the "Yankee Banker philosophy.") As soon as this service is available, we will publish an announcement to alert all our customers.

•AC•

C who's winning the race. Lattice C for Amiga.



Lattice C has long been recognized as the best C compiler. And now our new version 4.0 for Amiga™ increases our lead past the competition even further.

Ready, set, go. The new Lattice AmigaDOS C Compiler gives you faster, more efficient code generation and support for 16 or 32-bit integers. There's direct, in-line interface to all Amiga ROM functions with parameters passed in registers. What's more, the assembler is fully compatible with Amiga assembler syntax.

More great strides. The linker, Blink, has been significantly enhanced and provides true overlay support and interactive recovery from undefined symbols. And you'll have a faster compile and link cycle with support for pre-linking.

There's no contest. Standard benchmark studies show Lattice to be the superior C language development environment. With stats like these, it's no wonder that Commodore-Amiga has selected Lattice C as the official Amiga development language.

Going the distance. You'll experience unsurpassed power and flexibility when you choose from several cost-effective development packages. There is even a full range of supporting products, including a symbolic debugger, resource editor, utilities and specialized libraries.

You'll discover that your software purchase is backed by an excellent warranty and skilled technical support staff. You'll appreciate having access to LBBS—one of the world's first 9600 baud, 24-hour bulletin board services. And you'll be able to conference with other Lattice users through the Byte Information Exchange (BIX) network.

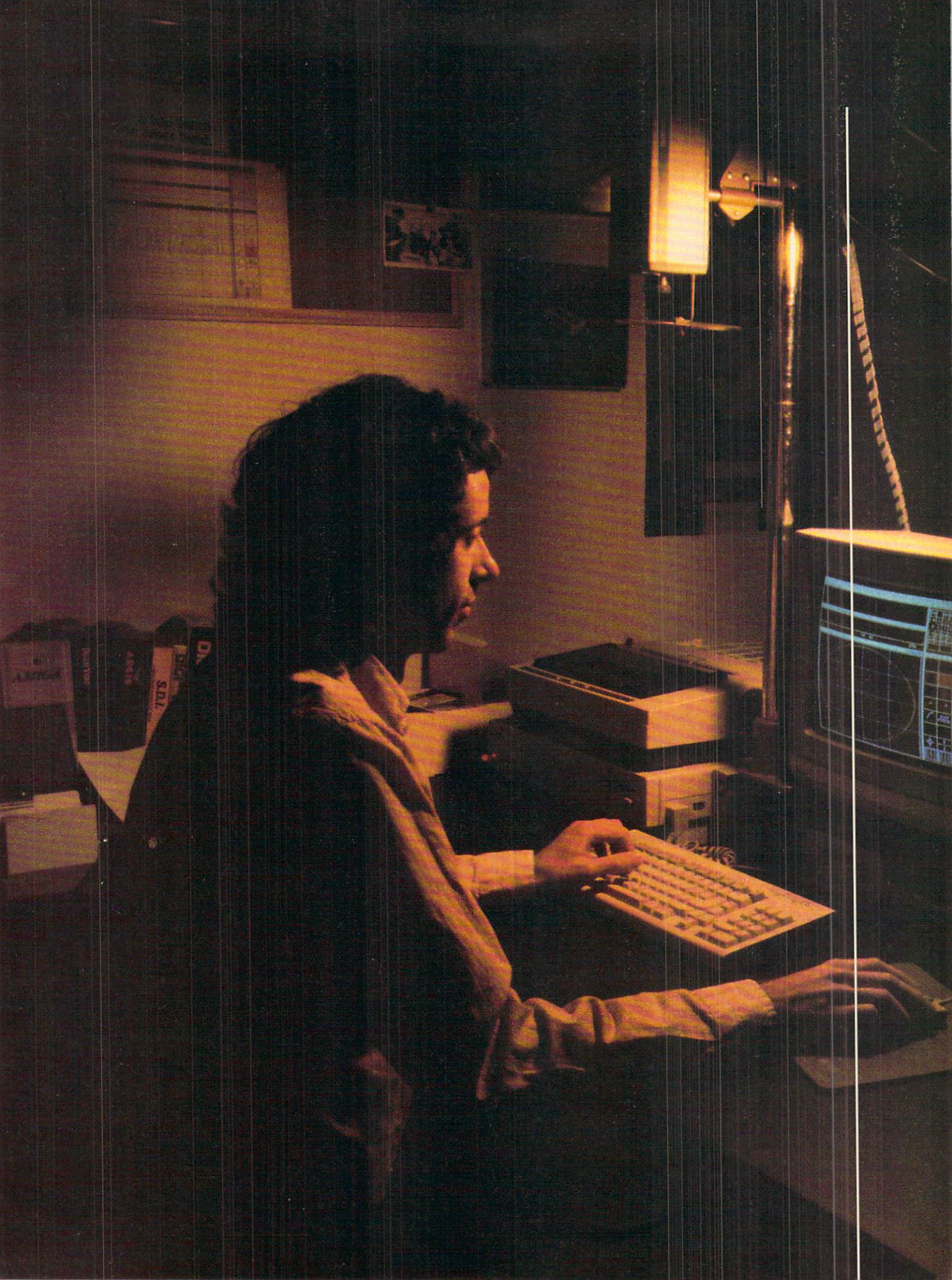
Cross the finish line. Order your copy of the Lattice AmigaDOS C Compiler today. We'll supply the speed. You bring the running shoes.

	Lattice® Version 4.0	Manx® Version 3.40
Dhrystone	1294 Dhrystones/second	1010 Dhrystones/second
Float	22.20 Secs. (IEEE Format) 10.16 Secs. (FFP Format)	98.85 Secs. (IEEE Format) 17.60 Secs. (FFP Format)
Savage (IEEE)	47.67 Secs./000000318 Accuracy	119.6 Secs./000109 Accuracy

Lattice is a registered trademark of Lattice Incorporated. Amiga is a trademark of Commodore-Amiga, Inc. Manx is a registered trademark of Manx Software Systems, Inc.


Lattice
Subsidiary of SAS Institute Inc.

Lattice, Incorporated
2500 S. Highland Avenue
Lombard, IL 60148
Phone: 800/533-3577
In Illinois: 312/916-1600





Beyond Laser Printing: Laser Light Shows with the Amiga

by Patrick Murphy, Art Director, Nightlight Laser Design Studio (Bix: Nightlight)

At Nightlight Laser Design Studio, we specialize in laser displays for theaters, art, and special events. To generate the projected images, we use Amiga computers to drive galvanometer scanners. Under computer control, mirrors on the scanners move the laser beam. With this technique, we can produce animations of virtually any size—from microscopic cartoons to writing on clouds.

This article is not a "how-to." Laser equipment is costly, and laser light shows are regulated by federal and state governments. This article describes, in a general way, the unusual technologies used in laser displays and how the Amiga plays a key role at Nightlight.

How Is a Laser Like a Stereo?

To produce a laser image, we use techniques first developed in the 70s which now have grown into a mature technology.

You start with a continuous-beam laser as the light source. Common lasers include low power helium-neon gas lasers, and higher power argon and krypton gas lasers. These produce different colors. Because of how the laser action is generated, helium-neon laser light is usually red. An argon laser produces superimposed green and blue beams; a krypton laser has red, yellow, green, and blue components in its "white" beam. Planetarium displays, such as Laser Images' "Laserium," use a single krypton beam split into those four primaries.

Tiny mirrors (.25 x .50 inches) mounted on oscillating shafts of precision galvanometers are used to move the laser beam around. One galvo is used for horizontal (X) motion, the other for vertical (Y). The beam reflects off one mirror, then off

(continued)

the other, and finally onto the projection surface. Often, a third galvo (or an electro-optic device) is used for intensity control, turning the beam on and off. A "Laserium" show uses at least 12 of these: X, Y, and intensity for each of the four colors.

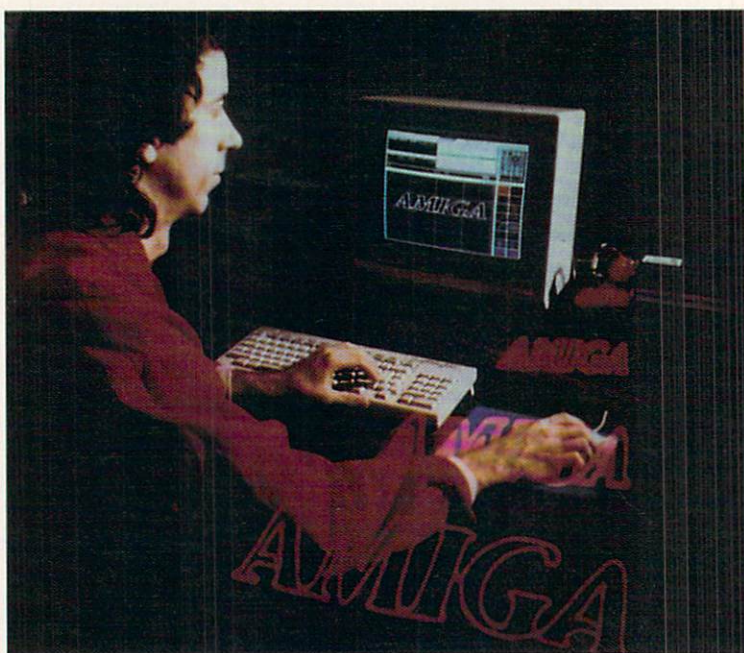
You can use any electrical signal—computer, synthesizer, or audio—to drive the galvos. Special amplifiers are used with the galvos to condition the signal and reduce inertia effects of the mirrors. There are two main types of scanner/amp pairs: position-detecting (PD) and non-PD. Position-detecting scanners have a feedback loop to ensure the most accurate tracking response. Non-PD scanners are usually used with abstract figures that do not have intricate detail. With the Amiga, PDs are a must.

A scanner system is analogous to a stereo system. Both use similar input signals. Both use amplifiers to condition the signals for the output transducers (galvos or speakers). With a stereo, you *hear* the electrical signal; with scanners, you *watch* it. Laser light shows are truly "music for the eyes."

Why Aren't Lasers Used More Often?

There are a few important limitations with laser systems.

First, you are dealing with light, which is massless and unaffected by electromagnetic forces. You use magnets to direct electron beams with great accuracy and "repeatability," as in



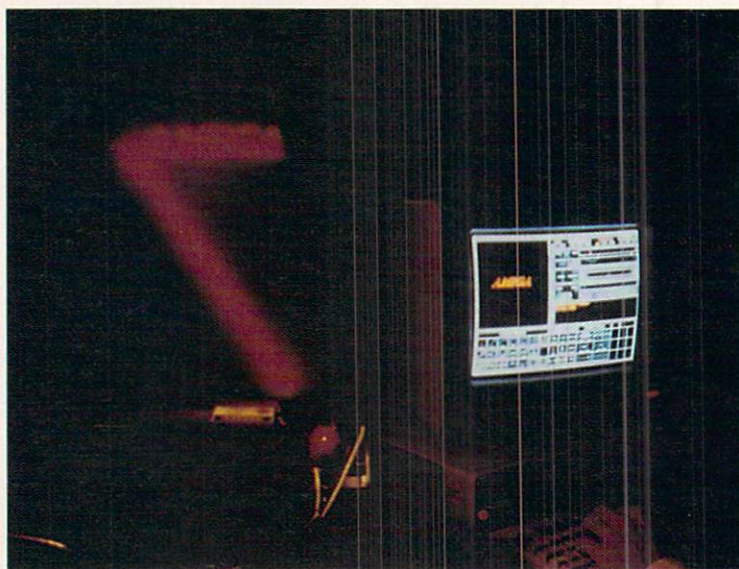
CRTs. With lasers, however, you need to move mirrors, which have relatively high inertias. Thus, scanners are useful only with signals from DC to about 3000 Hz. For best accuracy, most galvo scanning is done at 100 Hz or less. (Electro-optic crystals can move beams much faster, but the scan angle is limited to around 5 degrees.)

Second, you can't do detailed graphics with these mirrors. Designs cannot be too complex because the laser draws the entire figure, from first to last point, before the first points fade from your retina. Thus, most laser figures are simple cartoon-like outlines.

Third, you are dealing with a relatively weak light source. It takes a lot of electricity to

make a little laser light. A very powerful display-type laser would produce 20 watts of light. When concentrated into a small area, this is a lot. If you were to spread the entire 20 watts over a large rectangular surface, though (as you might in a laser-lit projection CRT), you have a relatively dim light source. Therefore, most laser displays use vector scanning of the concentrated beam. Raster scanning requires all the laser's light to be spread over the projection surface, even if only a small area is turned on.

Fourth, you are dealing with low-volume, high-precision devices. They cost many arms and many legs. Good PD scanners can cost over \$1000 per axis, with amp. A basic helium-neon for tiny shows is \$500. Our 2-color argon lasers for large shows each sell for over \$10,000. In addition, they can't be used just anywhere. Each needs 220 volts three-phase, and requires over 2 gallons of water per minute to cool the tube and power transistors. Think twice before



MicroBotics means Amiga-Power!™

Whichever Amiga you own—or plan to buy—we have the expansion you need

For the
Amiga 2000...

HardFrame/2000

Super Speed DMA SCSI Interface
If your application calls for super-speed uninterrupted access to your harddisk, HardFrame/2000 is your answer. This is a high-end, no holds barred SCSI interface that operates at bus speeds with zero wait states. With cable pinouts designed for compatibility with low cost Macintosh hard drives, one HardFrame/2000 can support up to seven devices. Word-length data transfer, FIFO buffering, true DMA, all mounted on a metal frame suitable for mounting standard SCSI 3.5" drives "hard-card" style (or, if you prefer, cable connected to a bay mounted or external disk). Available February / March. Suggested List price \$329.

SB2000 Adaptor Card StarBoard2 Portability

Large numbers of MicroBotics StarBoard2 owners have moved over to the A2000. To protect their investment in our technology we've made available a simple, low-cost adaptor card that permits the installation of a "de-cased" StarBoard2 inside the Amiga 2000 (in the first 100-pin slot). When adapted to the 2000, StarBoard2 is still fully functional autoconfiguring memory plus you get access to all the StarBoard2 MultiFunction options—StickyDisk, Math chip, parity or the new SCSI Module. Available now. Suggested list price is only \$49.95.

FivePower/2000 Eight Megs plus Hyper I/O

This is the dream card that all 2000 owners need—lots of memory and fast I/O. FivePower/2000 comes standard with SIMM sockets for EIGHT MEGABYTES of memory modules. This memory is user-installable to keep your expansion costs low. The RAM is organized in two autoconfiguring groups for maximum flexibility. Also included are two high speed, AT-style SERIAL PORTS jumper selectable as opto-isolated MIDI PORTS, and a socket and software for the 68881 MATH CHIP as a peripheral device (pioneered by MicroBotics on the StarBoard2 MultiFunction Module).

An optional daughterboard provides autobooting, zero wait-state, FIFO-buffered, DMA SCSI hard disk interfacing and software! FivePower/2000 is virtually everything you need on the Amiga 2000! Suggested list prices start at \$249 for 2 meg space with 0k installed plus two serial ports. Available March 88.

For the
Amiga 500...

M501 Memory+Clock Half a Meg at a Great Price!

As we are all coming to realize, a 1-megabyte Amiga (at least) is a necessity not an option. When you add the inboard 512k memory and clock module to your A500, make it a MicroBotics-brand, plug compatible work-alike. It uses the exact same kind of memory and the exact same clock and battery. And note that just like Commodore and unlike some third-party expansions, we use a long-lived rechargeable Ni-cad battery by Varta— which you'll never have to replace! Set the MicroBotics clock using the same software (on your WorkBench disk) as you use for the Commodore clock. What's the difference? You get to keep \$41 compared to the Commodore version. M501 has a suggested list price of only \$159.



MicroBotics, Inc.

Great Products Since the Amiga Was Born!

811 Alpha Drive, Suite 335
Richardson, Texas 75081
(214) 437-5330

SOLD ONLY THROUGH YOUR AMIGA DEALER

StarBoard2/500 Two Megs and a Choice of Modules

The premier memory expansion for the A1000 is now available on the A500. In a sleek, redesigned case with an independent power supply strong enough to power StarBoard2 and another A1000-style StarBoard2, all the power and flexibility of this great expansion device is available to you. Up to 2 megabytes of autoconfiguring, zero-wait state FastRAM, MultiFunction or SCSI module capability for either math chip/StickyDisk functions or fast SCSI harddisk interfacing. StarBoard2/500 also has a unique LED diagnostic/confidence light to indicate the powered up state of your Amiga and your expansion memory. Another A1000 style StarBoard2 can be connected to the expansion bus pass-UP (it exits through the top of the case) for a total of FOUR megabytes of memory and two modules. Suggested list price \$495 and up.

For the
Amiga 1000...

StarBoard2 The Expansion Product of Choice

The superb memory expansion for the Amiga 1000, still going strong! Up to 2 megabytes of zero-wait state, autoconfiguring FastRAM in a sleek, all steel Amiga-colored case plus the capability to accept either one of two daughterboard modules: the original MultiFunction Module or the brand new SCSI Module. StarBoard2 is powered by the bus (up to two StarBoard2's can be supported by the A1000) and passes it on. Available now; suggested list price \$495 and up.

MultiFunction Module High Tech at Low Cost

This "daughterboard" installs on any StarBoard2 (all three Amiga models). It features a socket and software to support the Motorola 68881 Math Chip as an I/O device (MicroBotics pioneered this approach on the Amiga—now directly supported in the math libraries in the new AmigaDOS1.3). StickyDisk gives you the most "bullet-proof" rebootable ram disk—its hardware write protection turns the whole device into a solid state, superspeed disk. Alternatively, parity checking of StarBoard2 memory can be enabled when extra parity RAM is installed. Finally, the MultiFunction Module carries an easy to use battery-backed clock to set your system time on start-up. Available now; suggested list price \$99.95.

StarDrive Module Speedy, Low-cost SCSI Interface

As an alternative to the MultiFunction Module, all models of StarBoard2 can accept this new hard disk interface. StarDrive affords you cost-effective, pseudo-DMA access to Macintosh compatible SCSI drives and other third-party SCSI devices. Fast, easy to install including driver software and disk diagnostics. StarDrive also has a battery backed clock to set your system time on boot-up. Available now. Suggested list price: \$129.95

MouseTime The Port Saving Clock

The easiest-to-use, most cost effective implementation of a battery-backed mouseport clock for the A1000. MouseTime passes the mouseport through for use with joysticks or other devices. Complete with installation software and WorkBench interface. Available now. Suggested list of \$39.95.

investing in laser display equipment. (However, the graphics output of signal sources is available to anyone with an X-Y oscilloscope or vector CRT. The laser-related components drive up the cost.)

Why Are Lasers Used at All?

So why use lasers? They have a few unique advantages.

The output of a laser scanner system can be as large as a mountain. Even the simplest system can go this big by using a large laser and beefing up the mirrors. Just as Cinerama is a different experience than TV, laser projections are very different from CRT displays.

Also, laser light has its special charms. Because of the way lasing action is produced, laser light shimmers and sparkles with spectrally pure colors. It adds effervescence to ordinary vector graphics.

Finally, laser displays can accomplish additional effects that are otherwise impossible. For example, the scanned laser can be directed over an audience to create Star Wars-like shooting beams. Or steady figures, such as circles, can be illuminated with smoke, so a tunnel of light is seen. Many other creative effects are available which add *oomph* to simple vector projections.

Subtle Nightlight Plugs

To see how laser graphics work in practice, let's take a brief look at some of Nightlight's projects. Most of these effects would be impossible to create with conventional projection CRTs.

Avant-garde theater director Peter Sellars wanted a spectacular effect for Anton Chekhov's "The Seagull." For the scene, Nightlight projected cones of laser light from behind actress Kelly McGillis and out over the Kennedy Center audience. As her character concluded her speech, the cones changed into darting shafts of laser light. The Washington Post described it as "...filling the Eisenhower stage with dancing beams of green light and clouds of smoke, which swirl about McGillis' immobile body like ocean waves."

Nightlight provided equipment for designs produced by MIT professor/artist Paul Earls. The laser was projected from a riverfront building over a plaza and onto a spray of water from a fireboat. The water screen was an appropriate projection surface for the images (and sounds) of sea creatures.

Nightlight did the opening ceremonies of the National Computer Graphics Association 1987 annual convention. The laser was projected onto a screen above the main entrance. It featured over 200 frames of animation, including a laser "Juggler," played live to two minutes of music.

One of Nightlight's mainstays is providing laser Tinkerbells for productions of "Peter Pan." We have a number of Amiga-scanner systems on the road, programmed specifically for the character's actions throughout the play. Tinkerbelle flies about the set, pulling the other characters' hair, and dying on cue. (The control screen has a slider marked "Death, Percent.")

One market that we shy away from is traditional rock shows. We stay away for technical, business, and philosophical reasons. Rock shows require more equipment, personnel, and enjoyment of Led Zeppelin than we can handle.

Why Use the Amiga?

To do complex, representational figures, you need a computer. Nearly any computer can be used to store and replay the numbers representing the figure's vertices. All you need is an area of memory which is read out to a digital-to-analog converter at a constant clock speed. For four long years, we used a 16K TRS-80 Model 1 with a custom interface box.

We chose the Amiga in 1986 after extensive research into other computers and laser systems. Many commercial systems were built around the Apple II, for example. Those systems had nice capabilities, but additional boards and custom software drove the total cost to around \$8,000. Other systems were little more than EPROMs with D/As.

The Amiga won because it does the most work with the least effort on our part. It was, at that time, the only computer with two separate audio channels (laser channels, for us). The audio hardware uses user-specified digitized waveforms, with variable speed readouts and waveform lengths. This is exactly what we need for laser control. For our purposes, we could use more outputs (for more scanners), but the base machine is a near-perfect laser machine.

Note: The Apple IIGS and Mac II now come with stereo. However, Apple's scheme uses multiplexing when reading out the waveform. I'll avoid getting technical and just say it's a typical clever Apple hack which, in this case, irretrievably muddies up those channels for laser use.

At Nightlight, we made one tiny Amiga hardware change. The normal audio output is monopolar and filtered (AC coupled). For lasers, monopolar

(continued on page 14)



Other Products from The Other Guys

REASON - a professional proofreading system used by universities and writers around the world to analyze and improve writing. (Has helped raise students grades when used faithfully.) \$395.00

OMEGA FILE - a *REAL* data base & mail merge \$79.99

PROMISE - the *BEST* high speed spell checker. (Even better than Zing!® Spell) \$49.99

KEEP-Trak GL - general ledger for home or business \$49.99

AMT - amortization program \$39.99

MATCH-IT - teaches shapes & colors (preschool) \$39.99

MATH-A-MAGICIAN - add, subtract, multiply & divide \$39.99

Call or write for more information.

SYNTHIA High Performance Digital Synthesizer

A state of the art music tool which will:

Create digital **IFF Instruments** for use with nearly all music programs!

Modifying existing **IFF Instruments**. Use **SYNTHIA** on digitized samples to add reverb, wow, and other enhancements.

SOMETHING FOR EVERYONE:

Additive Synthesis - a traditional method which can create almost any type of instrument.

Plucked String Synthesis - simulates plucked strings . . . right down to the 'pluck'.

Interpolative Synthesis - a method which introduces the natural imperfections found in instruments.

(Instruments such as brass, woodwinds, pianos, etc.)

Percussion - build your own drum set . . . create any drum you desire.

Subtractive Synthesis - a simple method of creating instruments.

Special Effects - includes filtering, amplification, phasing, waveshaping, amplitude modulation, real reverb, and . . .

IFF Music Player - powerful and compact. Now you can enjoy those songs that needed a memory expansion before! Up to 32 tracks and 32 IFF Instruments! Supports chords, ties, etc.

IS IT LIVE . . . OR IS IT SYNTHIA?

Synthia uses the latest technology to generate realistic sounding instruments and even the new families of instruments sound real. A real synthesizer on a real computer!

Why buy digitized instruments when you can **SYNTHIAsize** them?

Requires AMIGA 512K

\$99.99

Copyright©1987, THE OTHER GUYS Software • AMIGA is a registered trademark of Commodore Amiga



THE OTHER GUYS

55 North Main Street
Suite 301-D
PO Box H
Logan Utah 84321

(801) 753-7620
(800) 942-9402



means "gets bigger to the right and up, instead of from the middle out." AC coupled means "ugly rounded figures whose points all try to sink towards the center."

To get around this, we open up each laser Amiga. We pick off the audio signal near the Paula chip, add a few components, and install new RCA jacks above the existing ones. (You can always spot our Amigas in a crowd; they look quadrophonic-ready.)

This modification gives a second set of bipolar DC audio outputs for the lasers. Incidentally, sound from this set has much crisper highs, although you can also hear the whine which is normally eliminated by the anti-aliasing filter.

Why Use AmigaBASIC?

Our laser software is known as the MCP, named after the Master Control Program in the movie *Tron*. Like its namesake, it is powerful; but unlike the *Tron* MCP, it is very user-friendly. The MCP is intended to be the "Deluxe Paint of lasers," and it comes close. After years of hand-drawing figures on graph paper and typing in coordinates for the Model 1, it is a distinct pleasure to watch a laser beam track your mouse motions on a huge screen.

People using the MCP are surprised to learn that it is written in AmigaBASIC. Why? For a number of reasons, but mostly because our programmer (that's me) is too lazy to learn C right now. We do cheat a bit by PEEKing and POKEing at hardware addresses, instead of using the sound library. Between BASIC and the other Amiga libraries, the program does very well at its tasks.

There are only five aspects of the Amiga audio hardware we can control: 1) waveform data table; 2) waveform length; 3) output speed; 4) volume; and 5) channel assignment and attachment. The MCP allows fine control over these aspects, especially the waveform data.

How to Juggle X-Y Points

Here's a brief summary of how the MCP is used.

A laser figure is composed of a series of points on an X-Y grid. We use the 8-bit sound circuits for output, so our grid has 8-bit resolution: 256 x 256 possible points. Each figure is composed of as many as 512 points (X-Y pairs). These points can be drawn with DPaint-like tools, such as single point, line, box, circle, or symmetry. Any figure, or sub-group of points, can be moved, resized, or rotated. Sprites and bobs are used for slider gadgets and for identifying currently-selected points. Individual points can be inserted, deleted, or moved using a rubber band-like function.

We have up to 1024 different figures available at any time. You can copy figures (or sub-groups of points) from one location to another. For animation, you can mark off a sequence of figures to be interpolated or played back.

For example, students studying at Nightlight created a laser version of Eric Graham's "Juggler" less than a week after seeing the original. First, they created two figures—the body and the juggled balls. Next, key frames were derived. On the body, the arms became a sub-group and were rotated around the elbow and shoulder joints. Similar transforms

moved other body parts and the three balls. The computer then calculated interpolations between key frames to produce the individual frames of the animation. Finally, the body and the balls were merged into a single laser animation, moving eerily like the original "Juggler," but projected 12 feet high.

To do a show like the one done for the National Computer Graphics Association, we use DPaint to storyboard the graphics. We then create the necessary figures and animation sequences. After motion checks, we use a macro-type language with commands like "StartEndShowsecSparklesec (15,40,9,3)." This particular command shows an animation from figure 15 to figure 40 for 9 seconds plain, with a 3 second sparkle. The macros are assigned to keys, and the laserist "plays" the animations live to music.

Coming Really Soon...

We continue to improve the MCP. Some improvements come from the Amiga's capabilities. With genlock, we can trace figures directly over the televised image of a design. Others come as a result of add-on hardware. We are looking into A2000 boards to control four sets of scanners simultaneously (or we'll just network four A500s). We will also be adding sync-to-tape capability to allow totally automated music/laser playback. Still other improvements will develop as the software continues to grow in sophistication, or if our programmer ever learns C.

For Nightlight, the Amiga is an excellent choice, both as a development tool and as an everyday work-horse. It makes the type of laser displays we do affordable, easy, and exciting.

•AC•



AT LAST!

***...real-time, LIVE! video on your
Amiga's screen.***

- **True Color:** just as it comes from your video source: camera, VCR, TV, anything. Direct, moving, in your Amiga's memory...our patented technology.
- **Fast:** video images in black & white, 32-color, and 4,096-color HAM. See 15 new images every second in black & white, 12 in color, 4 in HAM.
- **Save:** moving video, play it back, use it in other programs. Unlimited stills, too.
- **Video Effects:** real-time mouse-controlled...posterization, fades, color-keying, strobes, more.
- **Roll Your Own:** programmer's video library, hardware documentation, examples in C, Basic.
- **\$295.** Immediate delivery. This is hot.

To order call toll-free anytime:
Nationwide: 800-452-4445, ext. 1156
California: 800-626-9541, ext. 1156

For more information, contact:

α^2

A-Squared Distributions Inc.

6114 La Salle Avenue, Suite 326
Oakland, California 94611
415-339-0339

The Ultimate Video Accessory

Part III

Making an exciting desktop video takes lots of planning and a variety of software for the widest selection of effects.

by Larry White

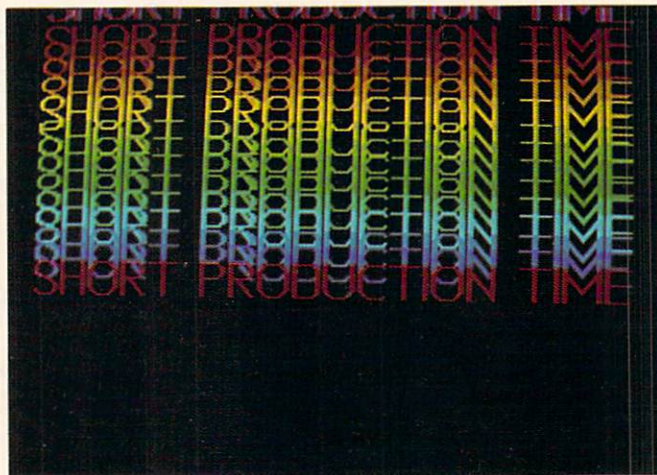
We've covered most of the basics in the first two parts of this series, so we should be ready to start making a video. The key to a good video is planning. Let's design a video that demonstrates many of the Amiga's amazing capabilities, and gives us a chance to try out many different software packages and a few interesting hardware devices along the way.

One of the best videos I've come across is the Amiga 500 "Test Flight." (You can borrow it from your Amiga dealer.) It's a remarkable 15-minute journey through the Amiga's capabilities. The production values are excellent, and almost every effect is within reach of your desktop video system. Watch this one closely before you make any videos of your own.

Generally, good video technique minimizes the use of special effects, to give maximum impact to the effects that *are* used. Since we want to try as many programs as possible, we'll create a special situation and break this "limited effects rule." As a useful exercise, let's pretend we're going into desktop video as a professional sideline. We need to produce a short commercial which demonstrates many of the techniques and special effects that exist in the Amiga video arsenal.

Video Considerations

Any screen you can create on the Amiga can be used in your video. However, if you're designing a screen specifically for a video, keep a few things in mind. First, your final video will not be quite as sharp as your RGB Amiga screens, so avoid small details which may be lost when the finished video is viewed on a less sophisticated television or monitor.



You should also keep a careful eye on the edge of your screen. Not all TVs and monitors are adjusted properly, and if you put important information too close to the top, bottom, or sides, it might be partially lost and illegible. If you're working with an Amiga 1080 monitor, leave at least a half-inch margin all around.

Don't completely ignore the margin you've created, though. Make sure your background graphics extend completely to all sides of the monitor screen. Your videos will have a boxy

look, with your images centered on the screen. Otherwise, make sure you capture the full video image by setting the software to "overscan" or full video. Most video-oriented Amiga programs offer this option, or will include it in an update—check with the manufacturer.

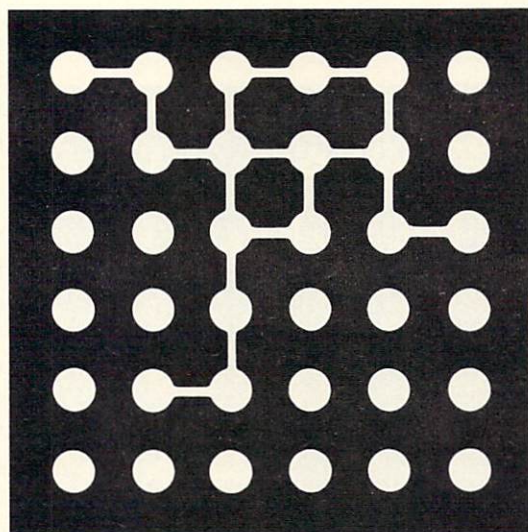
Another consideration is the video output from your Amiga to your video equipment. The Amiga 500 and 2000 offer only NTSC black-and-white or RGB video output. The Amiga 1000 does have a composite video output, but most users have been dissatisfied with the picture quality through this port.

Although several devices can produce high quality NTSC composite video from RGB, you'll probably want a genlock device. Since you'll be using your computer for other things,

or at least for designing your graphics, you'll want to use RGB video whenever you are not mixing with live video or recording. Unless you don't mind connecting and disconnecting devices constantly, make sure your genlock has an RGB passthrough.

Mimetics' Amigen is the first genlock available for all three Amiga models. Priced under \$200, this device currently stands alone in cost and compatibility. Several higher-priced genlocks are on the way, some with advanced features. We'll discuss them in due time.

(continued on page 18)



What's a Haicalc?

(pronounced hī-călc)

Haicalc™ is the best valued spreadsheet available for the Amiga. If you have ever had to project budgets, calculate expenses, or prepare a financial statement you need Haicalc. For only \$59.95 you receive a powerful program that fully utilizes the Amiga Intuition user interface.

You have the convenience of simple point & click selection of files under Workbench, pull down menu selections for commands, and economical use of your computer's memory -- program size is only 120k. You harness the power of multi-tasking and macros with

a maximum spreadsheet size of 9,000 x 9,000. Compare this to the competition. Haicalc is a powerful solution for a painless price.

SPECIFICATIONS

Maximum Spreadsheet Size — 9,000 rows by 9,000 columns*

Number Precision — Accurate to 8 digits

Interlace Display Ability — Up to 44 rows of data per window

Window Resolution — 1,000 by 1,000 pixels

Address Space — 18,000,000 cells*

Program Size — 120k

Sparse Matrix Allocation — Uses memory only when a cell contains data

Copy Protection — None

System Requirements — 512k Amiga, 1.2 Kickstart, Single floppy drive

*All capacities are subject to limits of system memory and storage.



HAITEX RESOURCES

208 Carrollton Park
Suite 1207
Carrollton, Texas 75006
(214) 241-8030

My Amigen is permanently connected to my A500. When I'm not using the computer for a video application, I have my monitor switched to RGB video. The Amigen has an RGB passthrough, but you'll need a separate 9-pin male-to-female cable. If you are using your computer this way, you must set the system to interlaced mode. If you don't make this adjustment, you may find some strange things, such as a flickering cursor and a fifty percent slower system clock which affects the operation of much software.

The Amiga 1000

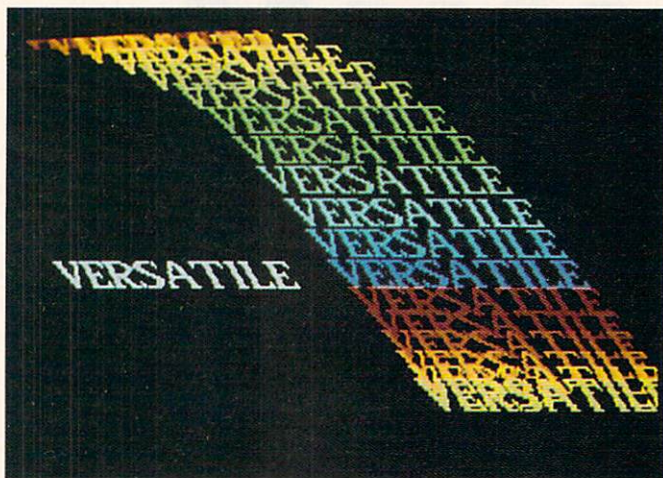
senses genlock and automatically sets interlaced mode. On the A2000 or A500, you'll need to set this manually. Amigen includes a program called Setlace, which you can install on your Workbench disk and fit into your automatic startup sequence.

While the Amigen is designed primarily to mix computer graphics with other video signals (synchronizing the computer's output to the external video), it also produces a quality NTSC composite video signal compatible with all monitors and VCRs designed for the U.S. (and a few foreign countries). You get this signal even if you don't have an external video source connected to the Amigen "video in" connector.

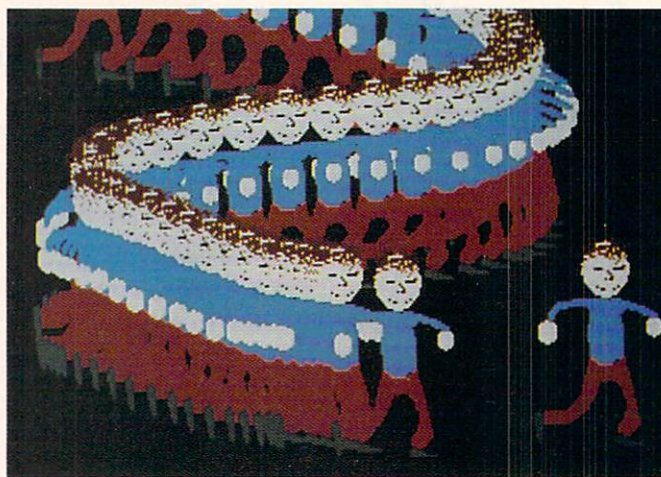
If you're planning to use genlock at all, you must know where the external video goes, so you can plan your graphics and animations carefully. You don't want any surprises when you start recording.

The genlock device replaces color register 0 with the external source. Many programs use black for color 0

as a default. In this case, any black area would be replaced with the external video. If you want to use black in your graphics (and not have it replaced by external video), you must assign an additional color register to black, effectively reducing your



available color palette by one. If you want to know which black areas are transparent and which are not, you can change register 0 to any color you won't need in the final graphic.



Unlimited Possibilities?

Of course, there are a few things you can't do on a simple Amiga desktop video system, particularly regarding the external video image. You may have seen TV commercials that shrink

the video image, then twist it and move the animated image across the screen. With the Amiga, unless you add some rather expensive video accessories, the external image must always be full screen (although much of it might be masked) and horizontal. Later I'll show how you can simulate some more advanced video effects with careful planning and controlled shooting.

The First Frames

Boot Deluxe Paint II or your favorite paint program. Using the "set page size" selection from the project menu, select "Full Video" to make sure you can use the full video screen. Any resolution mode can now be used, although you'll actually be using an interlaced output later on.

To reach the edges of the screen, scroll the screen with the arrow keys. If you want to view the full screen (in a slightly reduced format), select "show page" from the project menu (or press Shift-S).

Next, load the font directory and select a font for your titles. Keep in mind

that you want large crisp letters with ample contrast to the background. You want your titles to be clear and legible in the final video.

Try various combinations of text and graphics, keeping in mind that simpler is generally better. Save a few screens as IFF

files for next time, when we'll start animating the titles.

•AC•

[See the "Our First Desktop Video" Sidebar for more info. —Ed]

Hardware Suppliers:

Anakin Research, Inc.
100 Westmore Drive
Rexdale, Ontario, Canada
EasyI (Drawing Tablet)

Applied Visions, Inc.
Suite 2200 One Kendall Square
Cambridge, MA 02139
FutureSound (Sound digitizer)

Digital Creations
1333 Howe Ave. Ste. 208
Sacramento, CA 95825
Supergen (Video genlock with fade control)

Mimetics Corporation
16360 Stevens Canyon
Cupertino, CA 95014
Amigen (Video genlock)

New Tek
115 West Crane St.
Topeka, KS 66603
Digi-View (Video Digitizer)
Digi-Droid (Automated accessory for Digi-View)

SunRize Industries
3801 Old College Road
Bryan, TX 77801
Perfect Sound (Sound digitizer)
Perfect Vision (Video digitizer)

Software Suppliers:

Aegis Development
2210 Wilshire Blvd. Suite 277
Santa Monica, CA 94039
Animator, Images, Impact, Videoscape 3D
Aegis Titler

Brown-Wagh Associates
16795 Lark Ave. Suite 210
Los Gatos, CA 95030
Analyzel, TV*Text, TV*Show, Zuma Fonts

Byte by Byte
Arboretum Plaza II
9443 Capitol of Texas Hwy. N. Suite 150
Austin, TX 78759
Sculpt 3D, Animate 3D

Eagle Tree Software
PO Box 164
Hopwell, VA 23860
Butcher

Electronic Arts
1820 Gateway Dr.
San Mateo, CA 94404
Deluxe Video, Deluxe Paint, Deluxe Titler,
Deluxe Productions

Haitex Resources
208 Carrolton Park Suite 1207
Carrolton, TX 75006
Haicalc

Hash Enterprises
14201 SE 16th Circle
Vancouver, WA 98684
Animator: Apprentice, Animator Jr.

Impulse Inc.
6860 Shingle Creek Pkwy. #110
Minneapolis, MN 55430
Silver

Mimetics Corp.
16360 Stevens Canyon
Cupertino, CA 95014
Soundscape

MicroIllusions
17408 Chatsworth St.
Granada Hills, CA
Musix-X, Micro Midi, Micro SMPTE, Photon
Video

Micro Magic
Suite 320B 261 Hamilton Ave.
Palo Alto, CA 94301
Forms in Flight

Mindware International
Barrie Ontario, Canada
Pageflipper

New Tek
115 West Crane St.
Topeka, KS 66603
Digi-Paint

Par Software, Inc.
PO Box 1809
Vancouver, WA 98666
Express Paint

Progressive Peripherals & Software
464 Kalamath St.
Denver, CO 80204
Pixmate

PVS Publishing
3800 Botticelli, Suite 40
Lake Oswego, OR 97035
JDK Images CG1 (Character Generator)

Right Answers Group
Box 3699
Torrance, CA 90510
The Director

SunRize Industries
3801 Old College Rd.
Bryan, TX 77801
Perfect Sound, Perfect Vision, Studio Magic

Our First Desktop Video

As we try to get our fledgling desktop video company off the ground, we'll need to *show* what we can do for clients. A visually impressive, well organized video is our perfect marketing tool—it gives prospective customers a first-hand look at our video talent, while also outlining what we're all about.

Outlining the Adventure

Organizing our video is important, and creating an outline is a good way to start. The Amiga serves well in this area; you can use any word processor, or an "idea processor" (such as New Horizons' Flow), to write the outline.

Our outline should include all points and effects we wish to demonstrate concerning our fictional desktop video company, as well as suggestions for software and hardware options we want to try for each part. Once the outline is finished, it will be easy to create a storyboard and a script—the final steps before actually making animations and shooting live segments.

(continued)

Our outline should be dynamic. Even as I write this article, several new programs and a few hardware devices are on their way into the market. Whenever possible, I'll adjust our video to include the very latest software and hardware, and demonstrate the newest video features. I'll suggest several software solutions for various effects, and we'll work through the creative process together, step-by-step. It's impossible to be an expert with all the intricate functions in each software package. In some cases, we'll be experimenting and learning some techniques from the various programs.

If you haven't already done so, take a few minutes to read the outline. Notice that I've also included suggested running times for the various segments, keeping a general goal in mind of 5 to 10 minutes for the finished video.

The Storyboard

Now let's make a storyboard—a series of pictures that illustrates, in sequence, the various scenes that make up the video. Before we load Deluxe Paint II and start sketching, let's decide what to include in the video.

We'll begin with a title screen of our company's name. We can use either a static title or an animated title. A static display that suddenly comes to life and starts your viewer's journey into your Amiga desktop video creation is a nice twist. A moving object which stops to reveal the title is another impressive trick.

A list of the video applications to be covered in the video should come next. We'll repeat this list before each section, using color to highlight the specific topic being demonstrated.

Next, let's show why a client might want to use our desktop video services, rather the services of some more established video production

facility. Fly a few convincing answers across the screen: versatility, low cost, and relatively fast production time.

Because animated charts and graphs can make meetings more effective and are easily distributed to branch offices, business presentations could become a good profit center for your clients. We'll need to show a few charts and graphs in our video to get things rolling. Graphics alone, or overlaid on a live video image, are a great way to introduce new products to a sales staff or potential customers. Clients will see that, using similar techniques, they can demonstrate all their products and even produce a video instruction manual.

How about showing a video catalogue? This application can be particularly useful for real estate brokers. Video tours of houses for prospective clients or branch offices are ideal. Character generators (or titlers) can spell out the details and help maintain interest.

A simple cartoon can also have its place in the video. Besides having obvious entertainment value, a cartoon can tell the client's story in a light manner without using any costly special effects to demonstrate complicated devices.

A local band just starting out might be interested in producing a low-cost rock video. Desktop video techniques, including animation, might give them the extra edge.

Finally, you'll want to show some absolutely dazzling effects, like having your animations jump right from the screen. This fairly simple trick combines the best computer animation with good video technique—a perfect close for our video.

DESKTOP VIDEO DEMONSTRATION OUTLINE

- I. Setup and System Checks—
Place at head
of tape for checking quality,
adjusting monitor and volume.
 - A. Color Bars—Set contrast (picture),
color (hue) and brightness.
 1. Software: Deluxe Paint
 2. Time: 60 SECONDS
 - B. Musical tone—Set volume control
 1. Software: Instant Music,
Deluxe Music Construction Set
 2. Hardware: Microphone for Live
Audio.
 3. Time: 30 SECONDS
 - C. Blank Screen—provide timing
between tests and start of video.
 1. Software: Deluxe Paint
 2. Time: 30 SECONDS
-Note: Record pure black to hold
color burst and sync signals.
- II. Titles
 - A. Main Title
 1. Software: TV TEXT, Deluxe
Paint, JDK Images—CG1
 2. Time: 15 SECONDS
 - B. Introduction Screen—tell viewers why
desktop video might suit their
needs.
 1. Software: JDK IMAGES- CG1,
Aegis Titler, Deluxe Titler
 2. Time: 5 SECONDS
 - C. Menu Screen—build list of major uses
for desktop video, each of which
will be highlighted before
demonstration.
 1. Software: JDK IMAGES- CG1,
TV Text, Aegis Titler
 2. Time: 10 SECONDS
- III. Business Presentations
 - A. Graphs and Statistics
 1. Software: Haicalc, Analyze!,
Aegis Impact, The Director,
Deluxe Video
 2. Time: 10 Seconds
 - B. New Product Announcements/
Advertisements
 1. Software: Deluxe Titler,
Deluxe Video, Aegis Titler,
TV TEXT, Digi-View, Digi-Paint,
Pixmate, Butcher
 2. Hardware: Amigen, Digiview,
Digidroid, Perfect Vision
 8. Time: 30 SECONDS
 - B. Training and education
 1. Software: JDK IMAGES- CG1,

(continued on page 22)

**NEW
for the
AMIGA**

Precisely™

**The Word Processor
that doesn't try to be
a desktop publisher.**

Programs designed for graphic artists are difficult to learn, hard to use, tie up lots of memory, cost too much, and do not handle text very well. Graphics printing on a dot matrix or daisy wheel printer is very slow.

Introducing PRECISELY. The word processor optimized for people who work with words not pictures! Everything you expect in a word processor, such as:

- What you see is what you get (WYSIWYG)
- Very fast screen update/printer speed
- Easy to learn and simple to use
- Converts PaperClip, Pocket Writer and SpeedScript documents
- Online HELP • Mail Merge • Keyboard Macros
- Oops Key to undo mistakes
- Built in spelling checker coming soon

This is PRECISELY the word processor you've been looking for at PRECISELY the right price! Only \$79.95 (sometimes less is more). Add \$3.00 for shipping and handling, CA residents add 6% sales tax.

*Don't fumble around with your Amiga files.
Let **QUARTERBACK** manage your valuable data.
The Quarterback sneak scores every time!*

QUARTERBACK™

*Quarterback is a **FAST** Hard Disk to Floppy Backup Utility for the Commodore Amiga, featuring:*

- Fast backup - 20MB in less than an hour • Full/Subdirectory/Incremental backup
- Full/Subdirectory/Single File Restore • Automatic formatting of diskettes • Automatic catalog of files • Automatic diskette sequence numbering and checking
- Runs with Workbench or CLI • Includes/excludes files by name and/or wild card • Selects files by date • Selects files by Archive bit • Estimates number of floppies needed • Accepts batch command files • Validates writes • Restores original file date/time stamp • Prints backup/restore report • Allows easy access to subdirectories • Beeps for floppy change • Provides AmigaDOS access from within • Convenient/user friendly error recovery • Multitasking • No copy protection • Works with all AmigaDOS compatible hard disk drives

**You'll have fewer "time-outs" with
QUARTERBACK managing your file backups.**

Put Quarterback on your team for only \$69.95 plus \$3.00 for shipping and handling, CA residents add 6% sales tax.

TRANSFER FILES

TRANSFER C64/C128 files to and from your Amiga!

DISK-2-DISK reads your PaperClip, SpeedScript and Pocket Writer documents or other files on floppy disk directly into your Amiga. Transfers all file types. Use these transferred files with your favorite Amiga programs.

- Reads/writes 1541/4040 and 1570/1571 disk formats.
- Converts Commodore/PET ASCII to Amiga ASCII and vice versa.

**TRANSFER MS-DOS and ATARI ST files
to and from your Amiga!**

DOS-2-DOS reads Lotus 123 worksheets, wordprocessing documents or any other files on floppy disk directly into your Amiga for use with your favorite Amiga programs.

- Reads/writes both 5.25" AND 3.5" MS-DOS disks.
- Reads/writes 3.5" Atari ST diskettes (GEM format).
- Converts ASCII file line ending characters.

Disk-2-Disk requires the Amiga model 1020 5.25" disk drive. Dos-2-Dos runs on any standard Amiga. Disk-2-Disk \$49.95, Dos-2-Dos \$55.00. Add \$3.00 for shipping and handling, CA residents add 6% sales tax.



Central Coast Software™

268 Bowie Drive, Los Osos, California 93402 • (805) 528-4906



- Deluxe Paint, The Director
- 2. Hardware: Amigen, VCR controller
- 3. Time: 20 SECONDS
- C. Logos
- 1. Software: Videoscape 3D, Sculpt 3D, Animate 3D, Silver, Deluxe Paint
- 2. Time: 15 SECONDS
- D. Real Estate and Video Catalogue
- 1. Software: Digi-View, Digi-Paint, JDK Images- CG1
- 2. Hardware: Amigen, Digiview, VCR Controller
- 3. Time: 20 SECONDS
- IV. Animations
- A. Cartoons
- 1. Software: Deluxe Paint, Deluxe Video, Aegis Animator/Images, Animator Apprentice, Sculpt 3D/ Animate 3D, Pageflipper, The Director, Photon Video
- 2. Hardware: EasyI
- 3. Time: 20 SECONDS
- B. Animated Effects (added to live video)
- 1. Software: Aegis Animator,

- Silver, Deluxe Paint, Forms in Flight, Animator Apprentice
- 2. Hardware: EasyI, Amigen
- 3. Time: 15 SECONDS
- C. Music Videos
- 1. Software: Instant Music, Deluxe Music, Music-X, Soundscape
- 2. Hardware: Midi Interface, Midi Keyboard
- 3. Time: 30 SECONDS
- V. Post Production Effects
- A. Titles
- 1. Software: JDK Images- CG1, Deluxe Paint, Deluxe Video, Aegis Titler, Deluxe Titler, The Director
- 2. Hardware: Amigen, VCR Controller, FutureSound, Perfect sound
- 3. Time: 15 SECONDS
- B. Maps

- 1. Software: Deluxe Titler, Deluxe Paint, Digi-View, Digi-Paint, Pixmate, Deluxe Video
- 2. Hardware: Amigen, Digi-View
- 3. Time: 10 SECONDS
- VI. Closer
- A. Advanced Techniques—Multiple Animations
- 1. Software: The Director, Deluxe Paint, Animator Apprentice, Instant Music, Animate 3D, Videoscape 3D
- 2. Hardware: Perfect Vision, Supergen
- 3. Time: 25 Seconds

— Larry White

ADD TO THE POWER OF YOUR PROGRAMS WHILE YOU SAVE TIME AND MONEY!

CBTREE does it all! Your best value in a B+tree source!

Save programming time and effort.

You can develop exciting file access programs quickly and easily because CBTREE provides a simple but powerful program interface to all B+tree operations. Every aspect of CBTREE is covered thoroughly in the 70 page Users Manual with complete examples. Sample programs are provided on disk.

Gain flexibility in designing your applications.

CBTREE lets you use multiple keys, variable key lengths, concatenated keys, and any data record size and record length. You can customize the B+tree parameters using utilities provided.

Your programs will be using the most efficient searching techniques. CBTREE provides the fastest keyed file access performance, with multiple indexes in a single file and crash recovery utilities. CBTREE is a full function implementation of the industry standard B+tree access method and is proven in applications since 1984.

Access any record or group of records by:

- Get first
- Get previous
- Get less than
- Get greater than
- Get sequential block
- Get all partial matches
- Insert key and record
- Delete key and record
- Change record location
- Get last
- Get next
- Get less than or equal
- Get greater than or equal
- Get partial key match
- Get all keys and locations
- Insert key
- Delete key

Increase your implementation productivity.

CBTREE is over 6,000 lines of tightly written, commented C source code. The driver module is only 20K and links into your programs.

Port your applications to other machine environments.

The C source code that you receive can be compiled on all popular C compilers for the IBM PC and also under Unix, Xenix, and AmigaDos! No royalties on your applications that use CBTREE. CBTREE supports multi-user and network applications.

CBTREE IS TROUBLE-FREE, BUT IF YOU NEED HELP WE PROVIDE FREE PHONE SUPPORT.
ONE CALL GETS YOU THE ANSWER TO ANY QUESTION!

CBTREE compares favorably with other software selling at 2,3 and 4 times our price.

Sold on unconditional money-back guarantee.

YOU PAY ONLY \$99.00 - A MONEY-SAVING PRICE!

NOW! Variable length records.

TO ORDER OR FOR ADDITIONAL INFORMATION

CALL (800) 346-8038 or (703) 847-1743

OR WRITE



Peacock Systems, Inc., 2108-C Gallows Road, Vienna, VA 22180

ARAZOK'S TOMB

reviewed by Kenneth E. Schaefer

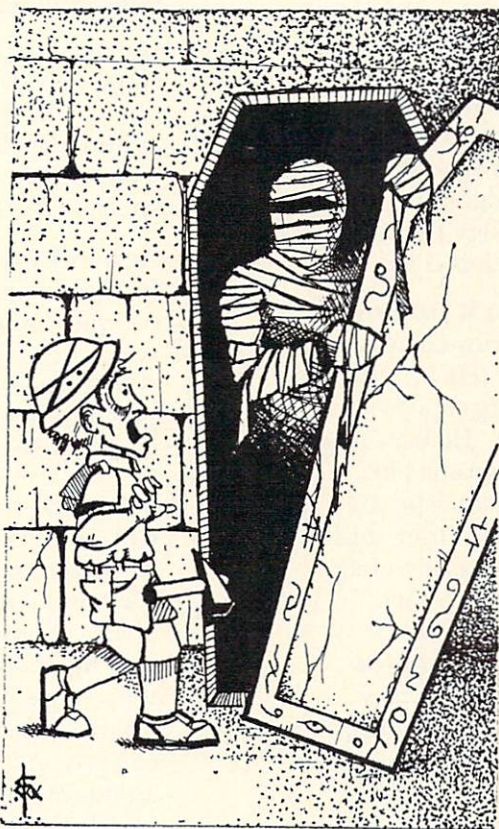
Archeology is a fascinating profession. What starts out as a simple dig in Scotland can suddenly become a terrifying adventure into the world of the occult. Accounting can't make such a claim; neither can teaching. You don't even have to be an archaeologist yourself. Another family member will do just fine for our purposes. After all, when they get into trouble, who are they going to call? Adventure-busters! And that means you.

Arazok's Tomb is Aegis's entry into the game market. It is an interesting first effort. If you had to label it, you would probably call it a "graphics text adventure." Although the game is not perfect, it has several features that surprise the player. The graphics, although only static screen images, are extremely well done. (The graphic style is reminiscent of Jim Sachs.)

Games of this sort can be judged in two separate, but interrelated, categories. Game mechanics measures how well the user-interface works, how consistent the game is, how smoothly and quickly it operates, how helpful the pictures are, and so on. The other category is plot, which measures how interesting the story is, how tough the puzzles are, whether the pictures create a mood, and so forth. Let's look at Arazok's Tomb from these two angles.

When you first boot, the startup-sequence asks for the date and time. "This is an unusual thing for a game to want to know," you think. You

had better answer the request, however. The program uses the entered time as the initialization for a real-time analog style clock on the screen. What purpose this serves is not clear. I've never needed it for the game itself. But maybe later on...



Besides the clock, several gadgets on the screen make your adventuring easier. A compass gadget allows you to click on the direction you want to go. There are even separate gadgets for up and down. A scroll bar allows you to scroll back the text that has gone off the screen. This feature is very helpful for remembering things

said earlier. Storage capacity is not infinite, however, so you should write down the really important information.

Standard Amiga menus help you along as well, freeing you from typing the standard requests of get, put, look, read, *et cetera*. This feature is not as useful as it could be, however, since you cannot point at an item on the screen and click it to select it. You can't menu-select "get" and then click on that big bag of gold, so you are never completely free from the keyboard. Whether this is a plus or a minus is mostly a matter of taste. Flexibility is always desirable, of course. Another menu selection is your inventory, which is limited to seven items. This option frees you from having to ask over and over during the game for a list of your possessions; that's a welcome help.

So much for mechanics—how is the plot? Well, this is where the game begins to sag. Arazok's Tomb is billed as an Adult Adventure, so I expected something a bit more racy than what I got. Maybe I've been hardened by horror movies, but a quick expression like, "The snake tears your throat out!" does not, in my mind, require the label "Adult." As far as I have played (I have not won yet), there has been nothing in the graphics to make me want to keep the under-eighteen crowd away. So why get our hopes up for a really "racy" game by using the label "Adult?" Marketing hype does not do a company any good in the long run.

(continued on page 27)

We Focus

In the growing video market one company leads the way in top-quality, low-cost, computer effects software. Aegis. You'll find our products working at every level in the video field. Professionals in television, video production, cable TV, industrial and government video, college, schools—you name it, we're there.



We're there because our products provide flexible technology that sparks the imagination. Technology that gets the job done.

You'll find us at Prism Graphics
For Jeff Bruette the Amiga spells success. He uses Aegis programs like VideoScape 3D, VideoTitler, and Animator to help him produce graphics and animations for television programs like *Max Headroom* and *Secrets and Mysteries*.

At MetaVision

When Theo Mayer and Peter Inova began work on a video project for Universal Studios tours they decided to try out an Amiga for a particular effect they wanted. It worked perfectly, and they saved thousands of dollars in post production fees. Next, they put Aegis products to work producing animations for a McGraw-Hill home



video. They were convinced, and the Amiga with Aegis software became an integral part of the studio.

At Master Communications
The publishers of

the popular *Board Sailing Year* annual video put Aegis software to work for their corporate clients. An Amiga 2000 is part of their video production house and they use products like VideoScape 3D, Animator and AudioMaster for special effects and logos.

And at Pixelight

Independent Amiga artist, Nick Poliko, uses Aegis products to create graphics and animations for such clients as CommCorp, Purolator, and Union Gas. They use his artwork for everything from public relations videos to attention getters at trade shows.

With Products for Graphics, Animation

We handle metamorphic, cel, and color cycling animation with Aegis Animator. Animations are created and edited in



an interactive environment allowing you to watch your animation as you build it. Winner of the 1986 CES Award of Excellence, Animator has already found thousands of uses around the world. Included in the \$139.95 price is the Images paint system.

Award winning in its own right Images is a lot of paint program for a small price. With color cycling, mirrors, air brush, pantograph, gradient fill, over 40

Aegis Animator, Images, VideoScape 3D, AudioMaster, Sonix, VideoTitler, and Impact are trademarks of Aegis Development, Inc.

ANIM is a trademark of Sparta/Aegis Development, Inc. Amiga is a registered trademark of Commodore-Amiga, Inc.

The AF8 Camcorder is a product of Sony Corporation and its use in this ad is not an endorsement by Sony of Aegis products.

on Video

other features, and a price of \$39.95, Images has the best price/performance ratio in paint packages.

Business graphics are a snap with Impact! Pie, bar, line, area, symbol, and scatter-gram charts are all easily created with every element of the chart from labels to axes available for editing. You also get a slide show generator with eight different dissolves and wipes. All for \$89.95.

For the advanced animator VideoScape 3D provides an environment rich in 3-dimensional capabilities. Object motion and metamorphosis, camera motion, light sources, IFF foregrounds and backgrounds, and the ability to create animations in the ANIM format are just a few of the features that make the \$199.95 price tag a great buy.

For titling you can't beat Aegis VideoTitrer. It supports all of the Amiga fonts as well as its own polytext fonts, works in four different resolutions and uses overscan. It has 20 different styles, works with IFF, uses halfbrite if available, and supports the ANIM format. Included in the amazing \$149.95 price is a slideshow generator that can mix ANIM animations with slides.



And Sound

Our video effects don't stop with great visuals. Programs like Sonix and AudioMaster take on the world of music and sound as well. Winner of a CES 1987 Award of Merit, Sonix lets you create your own instruments, compose music, and work with MIDI instruments. All for \$79.95.

We back up Sonix with AudioMaster, a digital sampling and editing program. It features interactive editing of the waveform and effects like echo, reverse, and low pass filtering. It also makes use of expansion memory (up to 9.5 MB) for extra long samples. Just \$59.95.

For more information and the dealer nearest you: (213) 392-9972 or to order direct: 1-800-345-9871.



AEGIS

2210 Wilshire Blvd., Suite 277
Santa Monica, CA 90403

HARD DRIVES

150 meg	\$1595.95
65 meg	\$ 949.95
40 meg	\$ 879.95
20 meg	\$ 659.95

Hard drive comes complete with SCSI controller, fan cooled power supply with case and a true SCSI hard disk (no interface boards).
Full ONE YEAR warranty!

**ASK ABOUT OUR REMOVABLE
MASS STORAGE DEVICES**

EXTERNAL 3.5 DRIVE ONLY \$ 159.95

CABLES: Serial or Parallel **\$ 9.95 each**

RAM **AMIGA 500** RAM

512K with CLOCK ..	\$ 129.95
1.5 meg INTERNAL .	\$ 389.95
2 meg EXTERNAL ...	\$ 449.95

AMIGA 1000

1.5 meg INTERNAL .	\$ 389.95
2 meg EXTERNAL ...	\$ 449.95

AMIGA 2000

2 meg INTERNAL ...	\$ 409.95
--------------------	-----------

ASK FOR OUR COMPLETE PRODUCT CATALOG

PIIONEER COMPUTING

-BLAZING NEW FRONTIERS IN PERSONAL COMPUTING-

2469 EAST 7000 South #200
Salt Lake City, UT 84121

(801) 942-1174

Some of the puzzles are just not very logical. I can remember the old days when the obstacles in a game were tough, but reasonable. Usually, you could be sure you would be able to solve at least a couple of the early, simple puzzles. These gave you the tools you needed to solve the later, more difficult ones. With more modern games, however, the puzzles seem to be designed illogically. Their solutions are not reasonable, and they start off hard, with no clues or hints. Of course, earlier games didn't offer hint books as separate purchases either. Once again, marketing darkens the bright sky of adventuring.

Another device that makes the game more difficult is limited vocabulary. In the beginning, finding just the right word was the whole challenge of text adventures. Public domain games like Adventure were that way. As time went on, though, the parser (the part of the program that breaks your text

input into its component words and analyzes them) became more powerful and understood some words as synonyms, so multiple words could have the same effect. Now, as if writers have run out of clever things to do, some games seem to be reverting to the old "guess what word I'm thinking of" style of puzzle. Why revert to more primitive parsers just to make the game more difficult? That's not the way to make games more challenging. This approach certainly doesn't give a very good impression of the creativity of the gaming industry.

My final complaint against Arazok's Tomb is not actually Aegis's fault at all. You will notice that the text scrolls very roughly. You may also notice a familiar looking screen in the background as the game boots up. Both of these characteristics stem from the fact that the game itself was written in AmigaBasic and compiled.

It is a testament to the utility and power of Microsoft's Basic for the Amiga that a game this good could be written in it. However, where AmigaBasic is weak, Arazok's Tomb is weak. We all know how poor AmigaBASIC's text rendering routines are. (Anyone who has used the editor that comes with AmigaBASIC knows what I mean.) Consequently, the text is written to its window in a rather choppy, jerky manner. This choppy-ness is only an esthetics problem, but you will notice it as you play.

Overall, I liked the game. The pictures are well drawn, and the plot is novel. I'm not sure how well players who don't have access to some outside help will fare. With games like this, being a member of a large and active user group or a Special Interest Group on a network is almost a requirement. Then again, who would journey into the sinister world of the occult alone?

•AC•

AiRT

REVIEW

by Steve Faiwiczewski

The ad for AiRT by PDJ Software reads as follows:

"AiRT is an icon based programming language that allows a programmer to create a usable program simply by selecting a series of pictures. AiRT was designed for non-programmers and those not willing to spend days or weeks developing a program using conventional language."

It sounded almost too good to be true, so I wanted to find out if the package lived up to its claims.

What You Get

The package contains one disk and a manual. The disk contains the AiRT editor, the compiler, a forms editor, a runtime support program, a print utility, a template "drawer," and one working example.

Using The Package

Using AiRT through Workbench is quite simple. To create a new program, just choose the "duplicate" option from the Workbench menu to create a duplicate of the template drawer. This sets up a directory with all the necessary support files. Once you have created a new drawer, open it to see the four icons inside: one each for the editor, the compiler, the print utility, and a log file created by the compiler.

You can run the editor by double-clicking its icon. The editor presents a map of all the program's "frames," and you must select the frame with which you want to work. A frame is

similar to a procedure in more conventional languages, such as Pascal and C. Once you click on the desired frame, the editor displays the frame's contents in the **edit work screen**. A frame consists of 78 cells. A cell is the smallest unit of execution, like a statement in other languages.

Cells are organized in three rows of 26 cells each, but the work screen can display only seven cells at a time. You can scroll through the visible cells with a sliding gadget at the top right of the screen. Figure 1 is a snapshot of an edit work screen with most of the cells already filled.

You place various icons into these cells, and they are executed at run time, left to right, top to bottom. The icons are selected at the bottom of the screen. Icons are set up in groups of six, and there are currently thirteen groups. Ninety-six icons are available in all, but four are reserved for future use. The grouping is logic-based—related icons are grouped together. Only one group is visible at any moment; a slider gadget scrolls through all available groups.

The editor lets you overwrite cells, modify cells which already contain icons, insert new cells between filled ones, and delete existing cells. Inserting and deleting cells really doesn't change the total number of cells in the program; the total is fixed at 78 (3 times 26). Insertion and deletion merely shift icons to the left or right (depending on whether you have done a deletion or an insertion).

You can fill a cell by clicking on the icon you want to place in the cell, and then clicking on the desired cell. The editor then presents a new screen, the **resolve decision screen**. A brief description of the selected icon is shown, as well as a list of all required parameters. Parameters are usually other cells, or **Fields**. Field types include: integer (small and large), float, string, file, form gadget, and form field. Figure 2 shows an example of the resolve decision screen.

Once the program is created and saved, the compiler is executed by double-clicking its icon. Compilation is fairly rapid and usually successful. Syntax errors are improbable because there is no syntax. The only problem I encountered was when I tried to refer to a form that didn't exist. (More about forms later.)

Once the program is compiled, an icon is created and displayed next time the drawer is opened. The program can now be executed at this point by simply clicking on its icon.

Available Icons

Icons are the heart of AiRT. They are the "actions," the objects that actually do something. Since there are over 90 icons, I'll mention just a few.

Icons are available to display IFF pictures, dismiss the program for a specified number of seconds (put it to sleep), and obtain the system date and time. Other icons display a requester and use the Amiga's built-in speech synthesizer. Some perform arithmetic;

(continued)

ATTENTION!

PROGRAMMERS & DEVELOPERS

The GS-1000 Graphics Library

- ★ A full featured INTERACTIVE Graphics Library maximizing the Amiga's™ powerful graphics capabilities!
- ★ One word commands eliminate hours and hours of programming time!
- ★ A complete library at your disposal, extensively documented and ready to use!
- ★ Built-in features include:

Device Independent Graphics • Multiple Coordinate Representations • Multiple Viewports and Screens • Zoom • Panning • Axial Scaling and Rotation • Rays • Palettes • Undo System w/Get Slate - Put Slate features • Color Mixing • Polymarkers • Brushes • Linestyles • Clipping • Interactive Text I/O • Debug Reporting facilities • Copy • Area Fills • Pattern Fills • PLUS standard features too extensive to list — over 90 commands in all •

PLUS A full featured Paint Program w/source to demonstrate the power of the Interactive Library.

Programmed in C, requires 512K memory.



Amiga is a registered trademark of Commodore-Amiga, Inc.

SEND CHECK OR
MONEY ORDER TO:

ONLY
\$49⁹⁵

Rittinghouse Software
Development Co.

P.O. Box 1272, Sioux Falls, SD 57101
VISA/MasterCard Customers Call:
Computer Business Center
(605) 342-6229 • Mon.-Fri. 9-5, Sat. 10-3
DEALERS INQUIRIES INVITED.

others manipulate disk files. Some handle gadgets, while others allow some graphic rendering. Icons are responsible for program flow control, and a branch icon allows branching to another cell, either within the same frame or to some other frame. Other flow-control icons perform test-and-branch actions (the branch is performed only if a condition is true). A Perform icon, very much like the PERFORM in COBOL or GOSUB in BASIC, is also included.

The Form Editor

The Form Editor is a really nice feature of AiRT. You can take an IFF picture and overlay gadgets and fields on top of it. You can turn any part of the picture into a boolean gadget, and you can create string fields (which behave as string gadgets) anywhere on the picture. These gadgets and fields are then usable from within an AiRT program. It is a pleasure to be able to design a data-entry screen using

DPaint, and then simply add the right fields and gadgets using the Form Editor. I wish this feature was readily available in other languages. Picture 3 is an IFF image with certain areas (highlighted) defined as gadgets.

The Manual

The 65-page manual consists of seven chapters and an appendix. Chapter 1 is a short introduction to programming. Chapter 2 describes AiRT and its uses. Chapter 3 discusses the editor, and chapter 4 covers the compiler. Chapter 5 covers the print utility, while chapter 6 explains use of the Forms Editor. Chapter 7 is a step-by-step example of creating, compiling, and running an AiRT program. The appendix lists all available icons.

The manual leaves a lot to be desired. The presentation is not coherent or logical. References are made to items

that are defined only at some later point, but you are expected to already know the definitions. The instructions lack direction and do not fit the target audience. The writing is overly simple at times, and too technical at others. Finally, there is no index, and some topics are not covered at all.

Good Points

AiRT presents a novel way of programming. The syntax-free method appeals to computer neophytes and those with syntax phobia. More experienced users may also find it useful because it's a very easy, fast way to do some "quick and dirty" programming. You could call AiRT a super-high level language—what you can accomplish with one icon would take many lines of code in a conventional language. The chance of bugs is also reduced because fewer program elements are used (fewer icons are used in AiRT than statements in other languages).

The Bad Points

There are quite a few:

A) The editors' user interface is weak. The package looks very user friendly at first, but a closer look reveals its deficiencies.

1. You cannot abort an edit session without saving changes. So, if you make a serious mistake while editing a program and want to abort the edit session, you must restore it to the way it was originally, or you lose the good version of the program.
2. You cannot save the current changes without exiting the editor. Introducing a minor change and then compiling and running a program to see how it looks is quite common. It's a real pain to restart the editor every time you need to change something.
3. Seeing 21 cells at a time can also be a nuisance. I would much rather have smaller cells and icons, so more can fit on the screen.

(continued on page 30)

InovaTools 1 from Inovatronics: The Right Tools for the Task

Developed by Todor Fay, author of the Soundscape system, these Amiga library functions add innovative, powerful features to your Intuition interfaces. Add a pop-up menu. Create a moveable gadget. Incorporate a knob gadget or a slick file requester. Remember, your interface creates that all-important first impression, so dress your software for the occasion!

InovaTools 1 includes:

- *File Requester:** This new file requester is small, fast, shows **VOLUME** names of all mounted volumes, performs filetype searches and keeps matched filenames stored for fast response.
- *List Handler:** These generic list-handling routines help control the display, selection and scrolling of a "list-in-a-box." Used in our new file requester, it makes list processing a breeze.
- *Drag Gadgets:** This new gadget type can be moved from one window to another, or anywhere in a specified display area without disturbing existing imagery. Great for "toolbox" programs.
- *Knob Gadgets:** Another completely new gadget type, this circular, easily manipulated, eye-catching gadget replaces standard proportional gadgets with accurate, calibrated, analog dials. Great for "control panel" gauges in games or other applications.
- *Palette Editor:** Incorporating the new "knob gadget," this palette editor is the epitome of "simple but accurate." Using both the red-green-blue and hue-saturation-intensity methods, you can set your palette by manipulating analog-type knobs. Changing any one knob results in the instant updating of all others affected.
- *Pop-up Menus:** This feature can pop-up a menu anywhere on the screen that you can put your pointer. Now, for the first time, you can have multiple menus in a single window for intelligently organized user-interfaces. Uses standard Intuition menu structures.

InovaTools 1 exists both in the form of system libraries that can be called from any language, and in linkable C code as well. And of course, when you want to include an InovaTools 1 gadget or pop-up menu in your program, you can use data generated by PowerWindows 2, our highly acclaimed, interactive screen/window/menu/gadget generator (version 2.5 and higher).

Voila! We've just solved your interface programming problems! Again.

only \$79.95

INOVATRONICS, Inc.

The REAL POWER in Power Programming.

11311 Stemmons Freeway/Suite 8
Dallas, TX 75229 214/241-9515

DYNAMIC DRUMS

The program that transforms your Amiga™
into a professional drum machine.

- Incredibly realistic sound
- Create your own studio-quality drum tracks
- Real or step time programming
- Graphic Editing
- Over 100 percussion samples included or use your own IFF samples
- Fully adjustable volume and tuning levels
- Randomizing options for a dynamic, human feel
- MIDI compatible

Requires 512K Amiga™
MI, FL, & CA add sales tax

DEALER INQUIRIES INVITED

Send Check or Money Order for \$79.95 to:



P.O. Box 438, St. Clair Shores, Michigan 48080

(313) 771-4465

Amiga is a trademark of Commodore-Amiga Inc.

4. The editor is very disk-based. When scrolling, many disk accesses really affect the editor's response time. I understand frequent disk access may be necessary on an Amiga with little memory (if you can call 512K little), but I have a 2.5 meg machine, so there's no reason to keep things on disk.

5. The editor cannot copy or move cells, making editing quite cumbersome. All text editors used with the more conventional languages have that ability, and it's sorely needed in AiRT.

B) The software does not work from the RAM disk. Since I have plenty of RAM, I wanted to load all the necessary files into the RAM disk, so editing and compiling would be much faster. Unfortunately, there is no documentation about doing so, and all my efforts brought the Guru. The situation has improved since I started using ASDG's FACC II, but PDJ should have provided this capability.

C) I came across a bug in the software. According to the documentation, a group of fields can be selected (as opposed to a single field) for some operations, such as writing to disk. I couldn't get this feature to work for form fields.

D) While AiRT's manual mentions "structured programming" on a few occasions, AiRT seems to go against modern-day structured programming standards.

1. AiRT's branching icons encourage sloppy code. They are really nothing more than GOTOs. No true structured programming constructs are included; nothing is available to match WHILE/DO, REPEAT/UNTIL or FOR loops of more conventional languages.

2. There are no local variables (fields) within a frame or a perform, and no parameter passing is allowed to frames

or performs. This means all fields are actually global variables—"evil" items in modern programming.

3. Since there are no local variables and no parameter passing, recursion is impossible. You may think recursion is an obscure, useless concept, but it does indicate the maturity of a language.

E) AiRT's most serious shortcoming is its lack of "expendability." Comparing AiRT to other languages is like comparing Chinese to English. In Chinese, each word and concept has a unique symbol (icon). A single icon represents a single concept.

In English, words are made up of letters, so a concept requires many symbols. However, a new concept or word in English requires just a new ordering of the existing symbols. No new symbols need be created.

In Chinese, a new icon must be created for each new word and concept. AiRT is like Chinese. New things require new icons. Unfortunately, only the authors of AiRT can create new things. This limitation makes AiRT quite restricted.

Summary

AiRT's paradigm of programming is innovative and powerful, with excellent potential. Unfortunately, most of that potential is not realized in the current implementation of AiRT, and I cannot recommend it for any serious work.

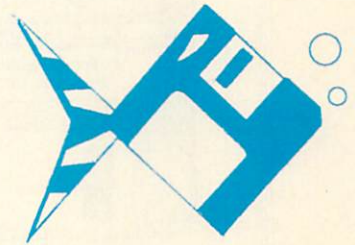
•AC•

AiRT \$64.95

PDJ Software
111 Thornwood Dr.
Marlton, NJ 08053
(609) 596-8991

Hooked on the Amiga with Fred Fish

by Ed Bercovitz



Among Amiga users, two individuals are considered "fathers" of the Amiga: Jay Miner, who headed hardware development, and R.J. Mical, who headed development of the operating system software. A third individual, whose contribution later in Amiga history was no less significant in ensuring the success of the machine, is Fred Fish. As you probably know, Mr. Fish is the originator and continuing force behind the "Fred Fish" public domain software collection.

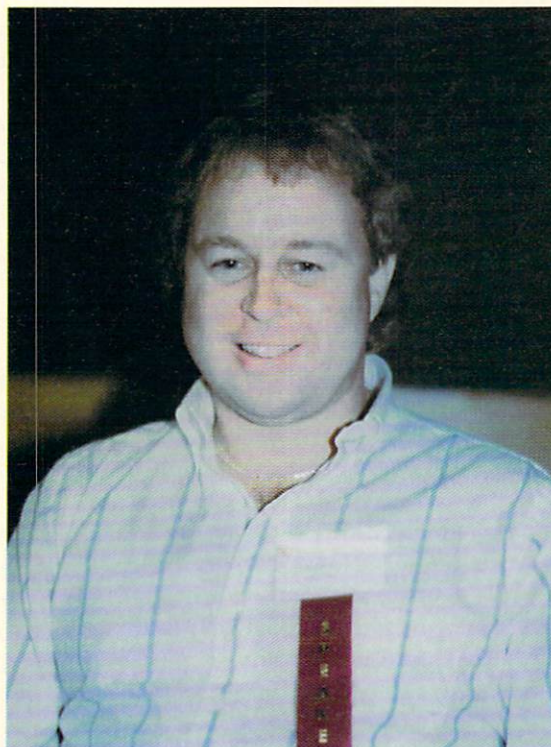
Before commercial software really started flowing, public domain material was the major source of software. More important, the circulation of public domain source code provided a synergistic impact for both amateur and professional software developers. This public domain spirit has typified the Amiga developer community. To a large extent, the medium of communication has been the Fred Fish disk collection.

Although the name is familiar, "Fred Fish the person" is largely unknown to Amiga users. *Amazing Computing* met with Fred Fish at the World of Commodore in Toronto.

AC: *How did you get into the business of producing public domain disks for the Amiga community?*

FF: When I got my machine just after Thanksgiving 1985, there was virtually no software anywhere. All I had was

the normal Commodore disks that came with the Amiga and a few demo disks that I had managed to con my dealer into letting me take home to play with. I said to myself, "I have to do *something* with this machine," so I started looking around to see what I



could find in terms of public domain software that I could port. I had also just bought one of the very early C compilers, an assembler, and a few other small tools, and I wanted to do something with them. At the time, I was working at a Unix company and had a fair amount of public domain software on disks that had come

through Usenet, so I started porting that. I had gotten two or three disks of useful stuff done when I heard about this user group called First Amiga Users Group which was meeting down in Belmont, in the San Francisco Bay area. So I went down there one day and I took along these disks that I had put together with the intention of distributing them. Well, the users there went kind of crazy because, of course, they didn't have much software either. That's basically how it all got started.

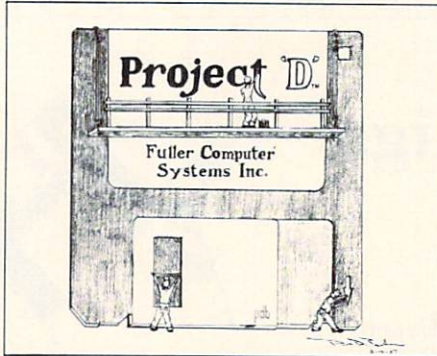
AC: *How much time do you spend each week working on the collection?*

FF: I do spend a lot of time working on the disks—probably more than I really should for my own long-term goals and professional growth.

It's interesting that some of the early disks were the ones that took the most time to put together. That was because, in the beginning, there were many programs that I sat down and went through line-by-line, and rewrote them to fix bugs. For example, there's a "make" utility in one of the first four disks that I went through, and reconstructed the whole thing from scratch. I built from what was there, but I really did reorganize all the routines. I probably spent a week just on that one program. Obviously, I can't do that anymore.

(continued)

INTRODUCING.....



An Evolution in Disk Utilities for Amiga™ Personal Computers!

**F
E
A
T
U
R
E
S**

- An easy to use, friendly and intuitive user interface.
- A powerful and fast disk backup tool that lets you make backups of your copy-protected Amiga software.
- A disk editing tool that lets you edit raw MFM tracks, AmigaDOS sectors and AmigaDOS files (automatically calculating new checksums).
- A disk cataloging tool that lets you maintain lists of your personal, public domain and commercial software.
- A unique backup tool for duplicating other disk formats including MS-DOS/PC-DOS and Atari ST.
- An easy to read, informative user manual is included.
- This product is not copy-protected in any way.

NOW SHIPPING!

\$49.95

EA.

Includes shipping and handling!
Arizona residents add 6.5% sales tax.

TO ORDER

Send check or money order to:
Fuller Computer Systems, Inc.
P.O. Box 9222
Mesa, Arizona 85204-0430
Or CALL (602) 835-5018

Amiga is a trademark of Commodore-Amiga, Inc.

Dealer Inquiries Invited

In general, programs that I redistribute now go out just as they come to me. I don't necessarily do anything to them other than possibly fix bugs that are reported to me between the time I receive it and when it is actually put on one of the disks. Averaged over a long time, I would say I spend 20 to 30 hours a week on the collection. The work goes in cycles. I try to make sure some disks come out at least every four to six weeks. There are usually a couple of weeks when I don't do anything other than fill orders; then there will be two weeks when I come home from work at 5 o'clock and work on the disks until I go to bed at midnight or 1 a.m., and I will also work all weekend. It does take a lot of my time. That's one of the things I'm short of nowadays, so that's probably why the rate of releases has slowed down somewhat.

I haven't charted how many of the disks are released [per] month, but one of the reasons I've been able to keep

up the amount of material is because people have started sending me more than they used to. So that greatly reduces the amount of time I have to spend to go out and track stuff down. Early on, I had to spend a lot of time actually going out and doing footwork to find material that was suitable and getting permission to include it, if necessary. Now I come home in the afternoon, and there's a whole pile of new disks that people have sent me to look at. Probably 70 to 80 percent of it isn't suitable for inclusion in the library, but there's some real gems that do turn up. It's really appreciated when someone sends me a program that is really useful and hasn't been seen before, it's all set to go, and all I have to do is to drop it onto a disk.

AC: How do you handle bug fixes and updates to existing programs in the library?

FF: This was something I had to deal with very early. I could have elected

to update the master disks with the new material, replacing the old material, but I figured that would be too confusing. Nobody would be able to figure out whether or not they had the latest copy of each disk, unless I went to some sort of numbering scheme that uniquely identified each disk update. Instead, I decided to treat updates and bug fix versions the same as new material and include them on a disk in the next batch to be released.

I wanted at some point to put together a "disk validation" program that would use a database of information about each disk, such as checksums for each file, which a user could use to make sure he had a complete and correct copy of each disk. Updating the master disks would have completely invalidated this idea, although I've never had time to do this anyway.

AC: There are currently 118 disks in the collection, totalling more than 100 megabytes. Reportedly, at one time, you had something like 300 megabytes' worth of programs and files. Does the difference of 200 megabytes represent material that wasn't included?

FF: It's far worse than that now. I probably have close to 2000 disks full of stuff. I've saved everything since I started doing this.

Yes, there's lots of disks—that's all I can say. I originally ordered a very nice disk box; it's about 18 inches deep and a couple of feet wide and six inches high and each one is supposed to hold 500 disks. Now I have a stack of four of those. In addition, I have a standard file cabinet in which a couple of drawers in there are practically full. The floppy is a nice little compact piece of media for hauling a fair amount of data around, but when you start talking about this amount of information, it's really inadequate.

I'm really looking forward to the day when all the material will fit on one optical disk; it will be much easier to keep organized. The way technology

is going these days, the amount of information available is so overwhelming. It's going to continue to be a problem, until we get some sort of compact way of dealing with it all. Having a stack of disks two feet high next to your Amiga isn't the best way to deal with it.

One of the problems I have now is that there is so much in the library that it's hard to keep track of it all. People come up to me all the time and ask me, "What disk is such and such on?" All I can say is, "Beats me!" When I was coming to the show today, I mentioned I had 8 new disks, and someone asked me what was on them—I replied, "Gee, I don't remember."

Of all the material that is submitted to me for possible inclusion, I try to keep at least one copy just for archival purposes, so if there is any question about where anything came from, I can always go back to the original disk somewhere in the pile of 2000 disks! Recently, when there was the unfortunate incident of some pirated commercial software making it onto one of my disks, I spent probably two days going through the massive pile of disks looking for a copy, trying to find out where the programs came from. It was never resolved. So I'm not sure how useful it is to keep everything.

And, of course, I have a lot of disks of stuff I've done on my own. A lot of it is digitized imagery. I have a Digi-View which I enjoy quite a bit, and I spend some time playing with that.

AC: Recently, I saw a reference to a collection called "The Best of the Fred Fish Collection." Is this something new you are distributing?

FF: The only one I am familiar with is a set that I put together for Commodore to release for their promotion of the 500 and user groups. I called this collection "The Goldfish Disks." I was really enthralled when I thought of that name. That set currently has only 3 disks, and unless I get a lot of

motivation to do some more, it may not go much further, at least for a while. The idea was to get some of the better and more useful pieces of software for the users out on a small number of disks, as a sample of all the public domain software that is available.

I'm sure there are a number of other people who have taken my collection and rearranged and extracted various pieces and so forth. Although I have no problem with that and encourage it, I don't necessarily want to get involved in it. I would prefer that they not call it the "Best of the Fred Fish Disks" directly, but there's not much I can do about it, if they choose to [use that name].

As you may be aware, it is possible, under copyright law, to copyright a collection. Very early on, I had to decide how to deal with that, and whether or not I in fact wanted to claim copyright on the collection. If I

had been going into it as a commercial venture with the idea of making my living off it, or even a substantial amount of secondary income, very definitely I would have copyrighted it. But my whole goal is simply to get the software out there. So that's why I decided to make it wide open, rather than claim any rights at all to it. But I would prefer that people didn't try and capitalize on my name, though.

AC: When you come out with a new disk, how many copies go out directly from you?

FF: At the present time, there are probably somewhere between 40 and 100 people or organizations that maintain a credit balance with me, and when new disks are released, they automatically get sent out to them. So this [is] the first tier of distribution. It's not large enough, so that it's overwhelming for me. If I had to deal with releasing 5000 disks, it would be

(continued)

Amiga-Tax II

Tax Planning & Preparation Software

Pay **\$69.95** once then receive
Yearly Updates for **ONLY \$29^{.95}**



CANADIAN VERSION

- All Schedules, All provinces, Gov't Approved
- PRINTS tax returns No manual transcribing!
- Save Time, Money, and Aggravation
- Useful all year for tax planning

Send \$10 for Mini Version & Save \$10 on final cost!

Datamax Research corp
Box 5000, Bradford, Ontario L3Z 2A6

a major chore. As it is now, I have my son do all the duplicating. I pay him 25 cents a disk for doing the duplicating and packaging, and he enjoys doing it.

Most of the people I distribute to directly are either geographically isolated from other users, [or] are what I would classify as "impatient collectors," [or] are user group librarians, or work in a store that sells the Amiga and maintain a library for their customers. Many stores can be persuaded to order the entire set of disks since they can then turn around and sell literally thousands of blank disks to their customers that want make copies from their store library. Sometimes I think I should have taken out stock in a disk manufacturing company before I started this!

AC: How can people obtain detailed information on which programs are currently in the library?

FF: I used to distribute a printed version of a library catalog that listed the contents of each disk. However, it is no longer cost effective to maintain and distribute a printed catalog, so I have gone to distributing a disk-based version. This catalog disk, library disk 0, is kept up to date at each release of a new batch of disks. Anyone can get a copy at any time by simply sending me a blank disk and sufficient return postage (currently 39 cents). There is no duplication charge for the catalog disk.

Once you've received the first catalog disk, you can repeat the process and specify that the disk be held until the next update. Unless you are on Usenet or BIX, this is probably the fastest way to find out about what is on the latest batch of disks. [Editor's note: Fred's postings on the contents of new disks are often reposted to other commercial networks and to local bulletin boards.] For disks that have been released for a few months, you can find an abbreviated catalog in the back of *Amazing Computing*.

AC: What sort of criteria do you use in selecting programs for your disks?

FF: First and foremost, it must be something that I'm allowed to include in the collection. So that leaves out commercial programs and pirated versions of commercial programs.

I also have a general bias towards material that comes with source code. Given everything else equal, if two programs come to me and one of them has source and the other doesn't, and there's only room for one on the disk, then, of course, the one with source gets put on. Other than that, there is just my general feeling as to how useful most people will find the program.

I also have a bias towards developers. The library was started initially to support the developer community and that's why the early disks were all CLI-oriented with no icons. So even if it's small, if a program does something useful for a developer, it will probably get included. I also recently have been trying to get programs which, if they do run from the Workbench, will run "nicely" in my environment (i.e., I don't have to change the disk name to make it work). I might go through a few gyrations trying to get it to run, but, in general, I like to include material that looks like somebody spent a little time on it, as opposed to something that somebody thought they would just throw out there.

Those are some of my criteria. Sometimes, when I'm trying to finish up a disk and I don't have a lot of material, something marginal may get included, not as a filler, but rather because it just happens to be the right size at the right time. So some of it is timing.

AC: Are there certain types of material you are less likely to include on your disks?

FF: I have some trouble with distributing programs written in BASIC for a couple of reasons. One is that I don't

"speak" any BASIC; I never learned it and never intend to. The other is that even though everybody has it, I can't put a version of Basic on a disk that you're going to give away. So that rules out having the program execute directly from my disk. People then have to spend some time constructing a bootable and "runnable" disk.

Some people ask me why isn't there more music stuff on my disks. There are a couple of reasons for that. One is that I don't think that there is any really good public domain or freely redistributable player for music programs, and everybody seems to have their own format. The other reason is the nature of music itself makes it hard to identify what is copyrighted and what isn't. I obviously couldn't distribute the digitized version of some Madonna song. Unless somebody gives me an original work, it's kind of hard to distribute anything that's really useful or interesting. So it's not that I have a bias against music, it's just hard for me to know what is distributable and what isn't. Finally, the other thing that makes music hard to distribute is that it takes a fair amount of disk space.

As for pictures, I haven't included anything that I've randomly downloaded. A number of artists have sent me samples of their work. That's great, and I like that. I try to pick and choose what looks to me to be the most interesting. I'm not a very artistic type, and perhaps some people aren't happy with my choices. I have artists who send me 2 or 3 disks of their work and ask me to distribute it. I look at it and and say, "It's OK," but I'm not sure if I was at the receiving end, I would appreciate receiving 3 disks of this guy's pictures. If there was an optical disk available and it only took two percent of the disk to put this stuff on, then there would be no problem. I would include it and maybe somebody would find it interesting.

AC: Have you ever thought of doing individual disks on particular themes?

FF: Putting a disk together on one theme would be a lot easier to do nowadays than it was in the past because, when I started, I pretty much put stuff on as it came to me. In some respects, that is what I still do. Although, if I have enough material and it's obviously related, I'll try to put it on a single disk. It wouldn't have been realistic in the 10s and 20s to say, "I need an art disk," and start collecting art, because it would probably take me six months to collect enough material to fill and release an art disk.

AC: Some of your disks contain shareware programs. How well do you think the shareware authors have done?

FF: At this point, I would have to discourage anyone from releasing a program as shareware. If the program is good enough that the developer feels that people would be willing to pay money for it, then by all means, try the commercial route. Put a small ad in an Amiga-oriented magazine and distribute it yourself for some small amount, like \$19.95 or something. Otherwise, just release it as public domain without restrictions. Very few programs have been successful as shareware products. Most people are simply too lazy or too busy to sit down and send the author the requested shareware fee. The motivation factor is the real problem with shareware, not the amount of the requested fee.

I should note that I no longer include shareware programs, unless they are sent to me directly by the author. I used to include shareware on disks for which I charged no duplication fee (every tenth disk), but I've discontinued this practice for various reasons. So if someone writes a program that they insist on releasing as shareware, and want it to appear in the library, they should send me a copy directly.

AC: What is the current situation regarding the pirated commercial programs inadvertently included in the collection?

FF: Basically, at this time, the whole matter is closed and has been laid to rest. I'm grateful that the company whose software was inadvertently distributed by me has chosen not to pursue the matter any further.

There was never any serious threat of legal action. The rumors about that [situation] greatly exceeded the facts of the case. There were rumors of the FBI getting involved and criminal charges. That is totally irrelevant in a case such as this, at least as far as my distribution was concerned. I doubt very much whether the FBI would be interested in any case that didn't involve at least hundreds of thousands of dollars. Certainly criminal action is only likely if someone knowingly and flagrantly violates the laws relating to copyright. The only possibility of

legal action that could have realistically been taken was some sort of civil action. Without jumping the gun by saying how I think it would have come out, it would have been a big mess for everybody concerned.

When the company contacted me, they wanted information on where my material came from and who I had sent it to; I gave them [that information]. I sent a recall letter to those I had directly sent the disk to. There was no shouting or jumping up and down. It worked out as well as could be expected under the circumstances.

There was really only one particular disk that had a problem. However, at this point, I have pulled 3 separate disks from the library. Disk number 57 was more of a judgment call. There was some software on it that was very similar to a commercial product, and I felt it really wasn't fair to continue to distribute the disk, since

THE DIRECTOR™



Professional display and animation language for the Amiga™

Envision a creative freedom you've only dreamed about. Imagine page flipping, color cycling, text generation, even IFF ANIM animations, all combined at the same time on the same screen. Now, from the simplest slideshow to the most sophisticated desktop video production, that dream comes true with the Director.

- Use any IFF images, any resolution, any number of colors
- Fades, Dissolves, Blits, Wipes, Stencils
- Page flip full or partial screens
- Preload images, fonts and sounds up to your memory limit
- Flexible script-based structure
- Basic-like vocabulary: For/Next, Gosub/Return, If/Else/Endif
- Arithmetic expressions, random number generator, variables
- Execute AmigaDOS commands from the script
- Text string and file input and output
- Keyboard and mouse interaction
- Digitized soundtrack module
- Supports HAM and overscan
- Supports IFF ANIM playback
- Built-in drawing commands
- No copy protection
- And much more . . .



The Right Answers Group
Department C
Box 3699
Torrance, CA 90510
(213) 325-1311

Amiga is a trademark of Commodore-Amiga, Inc.

DEMO DISKS \$10 each
Probe Sequence (512k)
RGB (1 meg)

\$69.95

Check or money order
payable to: Right Answers
Plus \$3 shipping and handling,
Calif. residents add 6.5% sales tax

(continued)

it was too closely related to what this company was marketing. Pulling the disk was done at my own initiative. No one came to me and asked me to remove it. In the other case, an individual was doing some work on contract, and the company he was working for felt he had taken advantage of some of their ideas in his public domain program. So rather than get in the middle of that, I decided to pull that disk also.

AC: Has other pirated stuff been sent to you for consideration?

FF: Nothing I can really point to and say for sure, "This is commercial." Nobody has been so obvious as to send me a copy of *Marble Madness* and say, "Here's a great game program I've written; please release it."

In the early days, it wasn't any real problem for me to keep track of what commercial software was out there because there was only a small amount of it. Now if you walk out here on the floor [of the show], it's obvious that there's a lot of it. If a program looks too polished or too professional, then I have to really question it. It is a real danger. You can be sure I do check out submitted material now a lot more closely than I used to.

AC: Recently, some programs include a notice that only you are allowed to distribute the program—an interesting phenomenon. What's your reaction to this?

FF: I'm not really sure what the motivation is of the people who include that sort of message. Obviously, if they send to me, I don't see how they can realistically expect that nobody else besides me is going to physically take the disk, make a copy of it, and give it to somebody else. They have to realize that I am the top of the pyramid for my series of disks. If it's OK for me to make a copy and send it to the people I deal with, I can't very well tell them, "sorry, you

can't redistribute this." I have to believe that they realize this and that permission for me to distribute the material implicitly grants permission for secondary distribution of the disk as released by me.

AC: Do you sense an increasing flow of public domain software in the Amiga world?

FF: Oh, yes, definitely. Only for the first six months was I able to keep up with everything that was available. After that, it was just overwhelming. Now I see probably no more than 10% or 20% of the stuff that is available. However, I generally see the better material, eventually.

AC: What is your hardware configuration?

FF: When I started doing the collection, I had a standard Amiga 1000 with 2 floppy drives. This was my basic configuration for almost a year.

The first piece of expansion hardware I bought was the ASDG Mini-rack C. I ran into Perry at a conference. It was the first time I had seen his hardware, and I was quite impressed with it. So I added the 2 meg memory expansion and then, later on, I expanded it to 8 meg. I then tried a hard drive and had [many] problems trying to get it to work with my memory expansion, mostly due to what I felt were problems with the hard disk controller design. I finally decided that the memory was more important than the disk drive, so I sold the drive off. Recently, I have been loaned a hard drive by a small company called Jefferson Enterprises. They make 20 meg drives that plug into the parallel port. That's what I'm currently using, and it has worked quite well.

I should be getting a 2000 soon. I'll probably keep the 1000 because, in some ways, I like the 1000 better, but I really need the expandability and the large disk storage capabilities of the

2000. The 2000 will be where I do all my disk work, but I suspect most of my everyday work, for whatever development I'm doing, will be done on the 1000.

AC: Are you still using your original floppy drives?

FF: I still have my original internal floppy; it has been used to copy all these disks over the last couple of years. When I did the user group disks for Commodore, they gave me a new expansion drive. I've got a new internal drive on order. I haven't received it yet, but it should be here soon. So I'm just now getting around to replacing my drives. There's been many thousands of disks that have gone through those drives and I'm surprised they've lasted this long. I think it's a good testament to their durability. In the last two years, these drives have had a lot more use than the average owner would ever see.

AC: What sort of software tools or utilities do you use in assembling the disks?

FF: Well, for an editor, I use MicroGNUemacs. I much prefer this [editor] over the other editors I've used. For compiling all the programs I get, it's pretty much a requirement that I have and use both Manx and Lattice C compilers. Each, of course, has its strengths and weaknesses. I have a Modula-2 compiler that I've never used and various assemblers that I very seldom use. For communications, I use Comm 1.34; it's done everything I want in a communication program, so I've seen no reason to go out and purchase a commercial one.

AC: Given the large amount and wide variety of public domain software distributed through your collection, has it ever been suggested that Fred Fish is the Amiga's worst enemy for commercial software development?

FF: I don't necessarily believe that, but I could see how credence might be given to that sort of idea. Who wants to spend 6 months or a year working on something great and wonderful, release it, and then find that 3 weeks later someone releases a public domain program that does seventy percent the same thing? It's really hard to say. It's interesting that I spend a lot of time using my Amiga and well over half the programs I use are public domain software.

AC: *Besides the Fish Disks, what else do you do in terms of program development?*

FF: The reason I bought the Amiga was because I wanted a good, fast machine with hardware support for graphics and multitasking. I wanted to try to do some digital signal processing software using multitasking with each little piece of the process as a task in itself, and then connect them all together with pipes.

So when I bought my Amiga, it was to do some development, but then I got side-tracked [by] the public domain disks. Up until this point, I haven't done any serious commercial work at all on the Amiga. That should be changing shortly.

I will be doing some more work on my hard disk backup program. This is a commercial product that has been reasonably successful in the Unix OEM market. I have a partner who does all the marketing of the program, and I do all the development work and technical support. This product worked out quite well in the Unix marketplace, and I'm looking forward to doing it on the Amiga. I had done a quick and dirty port about a year ago in order to verify that it was possible and to get a feeling of how much work would need to be done. Now I will be doing serious development, not only on the backup program, but also on some auxiliary programs to do filesystem and archive management operations. This is not a highly technical field, but it is something that is needed, and I feel this is

the right place and the right time to do it. This will be my first real commercial work for the Amiga.

AC: *Do you own any other PCs?*

FF: Yes, I have a very old Unix-based system I bought about 5 years ago. That was my first Unix experience. At the time I bought that system, I was working with Goodyear Aerospace and was working with PDP11s with the DEC operating systems, but I got interested in Unix. I couldn't convince anyone at the firm to buy a Unix machine, so I bought one myself. That machine is currently collecting dust in my spare bedroom, and I don't think I've turned it on in the last year.

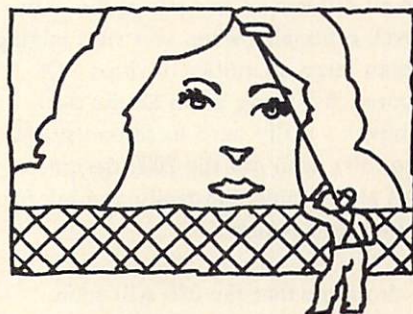
I also have a Mac II. I bought it primarily with the intent of running it as a Unix workstation—someday. I know a lot about how Unix was developed for that machine and I have the feeling it's going to be a nice machine to run Unix on, if and when it becomes available. I am not so interested in programming in the current Mac environment. Right now, the Mac II is relegated to keeping track of my customer list for my Amiga disks. I've had people comment that it seems like a very good application for a Mac. I don't have an IBM machine or a compatible, and I have no particular interest in that class of machine.

AC: *Obviously, the public domain project probably incurs more bills than it pays. What are you presently doing for a living?*

FF: I currently work with Motorola, doing language development tools such as compilers, assemblers, linkers, library support, and so forth. Some of it is for experimental architectures, and I find it interesting work. And, of course, it consumes a fair amount of my time.

(continued)

GET THE BIG PICTURE



"THE BIG PICTURE"

Print your IFF/HAM pictures in full color up to 104 feet x 104 feet. "Real" multiprocessing and overscan printed because printing is done from the file not the screen.

\$29.95 for the following printers in one package:

Canon PJI080A / Okidata 20 color Radio Shack Quadran Quadjet / Epson FX-80, FX-100, EX-800, EX-1000 / Fujitsu 1200, 2300, 2400 NEC P6, P7, CP6, CP7 / XEROX 4020

ATI-LINK

An IFF/HAM toTARGA file conversion program only \$100.00

Perferences Drives

Quadran Quadjet \$30.00 NEC CP7 \$50.00

Big Picture is a trademark of Lightning Publishing. Amiga is a registered trademark of Commodore Ltd. Targa is a registered trademark of AT & T

Lightning Publishing

1821 N. Ohio St., Arlington VA 22205
(703) 534-8030

Name _____
Address _____
City _____ State _____ Zip _____
Amount _____ Program _____

AC: Given your background and experience, do you feel there is a place for the Amiga in the high-end workstation market?

FF: Yes, definitely. I was personally rather disappointed when the 2000 came out that it wasn't a 68020-based machine. I think that the benefits of the 020 family clearly outweigh any small differences in price at the chip level, especially when you start talking about large quantities of chips. Of course, not being in on all the decisions, it's really hard to second guess the folks who did the 2000 design. But at this time, I'm really waiting for a high-end Amiga.

I also hope that the 030 will soon become available in sufficient quantities, so that when Commodore does a high end machine, they can go directly to that chip and if necessary, skip over the 020. I don't know what the pricing or availability of the 030 is at the moment. I like the idea that there is finally a 68XXX processor with a MMU (memory management unit) built in that system designers can't leave out. I think that is probably one of the reasons why the Intel 80286 and 386 have a bigger share of the Unix market for low end machines, because it's not possible for the people designing the hardware to leave the MMU out.

Therefore, by default, all machines out there with those chips are capable of running some type of Unix system. I think it's been fairly common knowledge in the Amiga development community over the past year that Commodore wants to make Unix available on the Amiga. However, I have no inside information on how well integrated Unix will be with the rest of the machine or AmigaDOS, or the backward compatibility with existing programs.

If you look where everybody else is headed--Apple now has the Mac II, and they've probably had Unix running on internally within the company for a couple of years or so,

even though it still isn't out. I'm sure Atari has Unix in mind. Steve Job's new venture, NEXT Inc., from all rumors I hear, will be an 030 based Unix workstation at high-end Amiga level pricing. The Amiga needs a high level model to complement the line.

AC: Do you see any reason why Commodore might not want to release high-end Amigas?

FF: I don't see anything other than price differential that might be holding Commodore up from putting an 020 or an 030 in an Amiga and making it a standard product. With a manufacturer installed 020, you would typically see a 3 to 4 times speedup; with an 030, it could be as much as an 8 times speedup. I would buy an Amiga that is 8 times faster than my current machine, even though it costs twice as much. That's me personally and it may not be representative of the market as a whole. However, the Amiga started originally as a hacker's machine, and that speedup allure is very hard for a hacker to resist. It may not be the biggest part of the market, but I'm sure they would sell enough machines to make it worthwhile.

Of course there might be some compatibility problems. Everybody is fighting that right now. The Mac family is finding compatibility problems between the Mac II and standard Macs with 68020 add-ons because of programmers who won't obey the rules and want to diddle the hardware directly, or they want copy protection to work in a certain way. I do hate to see that happening on the Amiga with programmers writing directly to the hardware. There is a fear at Commodore that if that situation is allowed to happen, they will be locked into a certain architecture and they won't be able to grow in the future. And of course the people that lose in the end are the consumers, since they don't really understand all the issues. If they buy a program that doesn't work on their machine it's hard for Commo-

dore to point the finger of blame at the software developers.

AC: *So what do you see yourself doing in five years, beyond releasing Fish Disk Number 900?*

FF: Gee, that's a hard one. I have so many interests right now that I almost feel schizophrenic that I'm doing too much at once with too many irons in the fire. My guess is that probably, within a year or so, I may burn out on doing the public domain disks. At that point, I may go into "hibernation" as far as the public world is concerned and start investigating the type of software I described earlier as the reason I bought my Amiga. My background is mostly engineering and math, and I feel I'm getting very rusty in my field. So I may go back for an advanced degree. It's almost embarrassing to say I don't really know where I'm going to be in five years. A lot depends on which way the chips fall. That's true, generally, of what happens in the world. I just wish I had a lot more time to spend getting back into the theoretical aspect of the sciences.

AC: *Have you thought about what would happen to the Fred Fish disks if you personally gave up producing them?*

FF: I haven't thought too much about that yet, but I suppose I would probably continue to make existing disks available to people, even if I wasn't producing new ones. At some point, I might just want to turn the whole thing over to some organization that wanted to continue to support it and continue the series.

AC: *Well, I'm sure all Amiga users hope that day will be later rather than sooner; when it does happen, it will mean the end of a major era for the Amiga.*

•AC•

Photographic Quality Reproduction with the Amiga and Digi-View

by Stephan Lebens

New Tek's Digi-View video digitizer is my most versatile Amiga peripheral. From the beginning, I have been able to produce stunning on-screen graphics with Digi-View. Of course, generating amazing screen displays is relatively easy with the Amiga; it's achieving professional quality output that is difficult. In this article, I'll describe how to produce high quality black and white hardcopy, using Digi-View and a Postscript printer.

First, a little background. The Amiga is capable of displaying and working with 16 discrete shades of gray (or grayscales), at a maximum screen resolution of just over 60 dots per inch (DPI). This figure is derived from dividing 640 pixels/10 inch screen area. This figure doesn't include overscan. It would seem that all you need to do is send this screen to a 300 DPI Laser printer for perfect results. Unfortunately, this is not the case. Even if you could somehow translate the 60 DPI bitmap to a 300 DPI bitmap, the number of grayscales would still be a problem. Compare a 16 color picture with a 4096 Ham image. To obtain this near-photographic quality with your hardcopy, you must dramatically increase the number of grayscales.

This is where Digi-View really shines! The hardware is capable of capturing 128 grayscales of information. This data is then processed for brightness and contrast, and reduced to the 16 grayscales with which the Amiga can work.

Fortunately, Digi-View designer, Tim Jennison, had the foresight to make the initial Image Processing file (IP) easily available. Simply set up Digi-View normally, without filters. Load the software and select either 320*200 color or 640*400 color. Select the menu item for color and click on the "BW(Black/White)" bullet. Now digitize from the Red menu selection. When the digitization is complete, press the F1 key. This will bring up a standard requester asking you to name the IP file and specify a path where it can be stored. A lores IP file is 64K; a hires IP file is 256K. You won't be able to use the hires IP file unless you have expansion RAM (as you'll see later).

Before I describe what you do with this rather large IP file, I must jump ahead to discuss the laser printer for final hardcopy. The only printer that works easily with this type of image file is a Postscript controlled Laser printer. A Postscript printer is a really a printer and a computer rolled into one package. It has at least 1.5 meg of RAM and the whole show is run by the same type of microprocessor that runs the Amiga. You'll see why all this computing power is needed later on.

Postscript printers range from the popular 300 DPI LaserWriter, all the way up to the 2450 DPI Linotronic series. More important than resolution is the number of grayscales these machines are capable of rendering. The LaserWriter can produce just over 40 grayscales, while the Linotronic 300 can show an amazing 256 grayscales.

Back to our binary IP file. The only remaining hurdle is converting the binary file into a workable format for Postscript. Postscript prefers bitmapped files in Hexadecimal (base 16) notation, rather than in Binary (base 2).

AmigaDOS handles this file conversion easily. From the CLI, use the "Type" command with the OPT H switch. Normally, the "Type" command displays its output to the screen. Example:

Line Number	Hex-Data	Hex-Data	Hex-Data	Hex-Data	ASCII-EQUIVAL.
0001	00000095	49FAFFEE	286CFFFC	2F0C2F02I...(l.etc.
0002	D9CCD9CC	200C6714	202C0004	E580B2B4g. ,.etc.

Instead of displaying the converted Hex file to the screen, you can redirect the output to a new file.

Example: "Type IPfilename opt h to df1:newfilename." (This assumes you have the IP file in the root directory of DF0: and want to send it to the root directory of DF1:.)

Now comes the part I mentioned earlier about expansion RAM. The lores IP file now weighs in at a hefty 248K; the hires IP file is a gargantuan 992K. Obviously, you cannot convert the hires IP file to Hex without expansion RAM because it won't fit on a disk.

(continued)

LaserWriter
(320 x 200) IP File



Original Artwork

(Taken from the cover
of the
*PostScript Language
Journal*, June 1987
Vol. 1, No. 2)

Linotronic 100
(640 x 400) IP File



Now comes the hard part. Postscript wants only the actual Hex data, meaning the line numbers and the ASCII equivalent text must be stripped from the file. (I am pleading with you programmers out there to produce a utility to strip this garbage. Please send me a copy when you're finished!) You can use a word processor which can import text-only files and has a Macro capability. I have been using WordPerfect. It easily handles the 992K hires IP file.

Develop a macro with the delete key and the cursor down key to trash the line numbers. Make another macro to delete the ASCII text. If you use WordPerfect, keep the macro around 100 lines in length; otherwise it can't store the macro in RAM, and you'll have a disk access each time the macro is invoked. Keep the macro name to one letter because you must type in a requester each time you call the macro. (You can't assign the macro to RAM because you need to specify the full pathname at each requester prompt.) I've managed to automate the process a bit by using Timesaver's macro functions, but sometimes it runs a bit too fast for WordPerfect! When you are done, remember to save the file as text-only.

While you're still in your editor, add a few lines of Postscript code to the IP file. This code causes the actual transformation of the image data to the printed page. This code appears at the top of the file.

Comments for other programs that might use this data:

```
%!PS-Adobe-1.0
%%Creator: Your Name
%%Title: Document Name
```

```
/picstr 320 string def
This would be 640 if you use a hires IP file.
```

```
384 288 scale
These numbers are the actual image size on paper. 72 are equal to 1 inch. Imagine an X Y grid. Change these to whatever suits you.
```

```
320 200 8
This figure means our image is 320 cells wide and 200 cells high. Each cell contains 8 bits of data (256 grayscales). I know this is a lie (we have 7 bits-128 grayscales), but the machine doesn't seem to mind! (Remember to change to 640 400 8 if you are using a hires IP file.)
```

```
(320 0 0 -200 0 200)
Here is more scanner info. This tells Postscript that Digi-View scans from bottom to top. Change the numbers if you are using the hires IP file.
```

```
{currentfile picstr readhexstring pop}
```

*For a detailed explanation image, see the Postscript Manual. Now comes all of that Hex data we have.

At the very end of the file, add the "showpage" Operator.

Remember to save this file as text-only. Strip out all of my comments. Just press a carriage return after entering every line, and you'll have no problems. Don't forget the "showpage" operator at the end of the file. For those of you who are interested, you can actually convert this image into Encapsulated Postscript. You could then use this file in any Mac or IBM (and Amiga soon, I hope) that supports EPSF. Simply change the first line to read:

```
%!PS-Adobe-2.0 EPSF-1.2
```

```
... and add the line
```

```
%%BoundingBox: xyxy
```

... to specify the bottom and top right hand corner of original graphic.

That completes the production phase. If you don't own a Postscript printer, you need to get this file over to your neighborhood Mac-based print shop. There are hundreds of these shops across the country. There are also many professional Linotronic-based production houses which can accommodate you. Many of these places offer Bulletin Board services for file downloading. Many have an IBM machine hooked to a Mac through the Tops network, so just format a Disk under MS-DOS on your 5 1/4 inch drive and copy your data to that disk (for lores data only). A few production houses may even take your Amiga disks directly!

Your first laser printout will make it all seem worthwhile. Pay the ten dollars and crank one out on a Linotronic. One look and you'll be hooked. Offer to do some digitizing for the print shop—it's the only way you'll be able to finance your new habit!

•AC•

The author welcomes all correspondence concerning this topic. I am particularly interested in using Digi-View as the front end in a color separation process. Please address all correspondence to the author care of this magazine.

Correspondence with the author is also available through the National Independent Postscript Support Board (409) 244-4704. Address to Slebens.

BALANCING YOUR CHECKBOOK WITH WORDPERFECT MATH & MACROS

Pay to
the order of

Memo

Steve Hull

GENie: LightRaider

By People Link: St.Ephen

One of the most frustrating things about reviewing productivity software is deciding which features to cover. WordPerfect is a perfect example of this. Ostensibly a word processing program, it has nearly two hundred functions, many of which are outside the scope of conventional word processing. Deciding exactly what to cover in a review and what to "let go" is a question I have not yet answered to my own satisfaction.

In my review of WordPerfect (AC V 2.12), I was especially frustrated because I was not able to spend more time on two capabilities that really set WordPerfect apart from its competition: the program's math and macro functions. With WordPerfect's math functions, you can compose mail-merged invoices in which WordPerfect calculates quantities and prices (including sales tax) and integrates the results within the document. Its macro functions allow you to compress pages of instructions into a two-keystroke combination. Put the two functions together, and you can literally turn WordPerfect into a spreadsheet.

At the risk of losing all credibility at the outset, I need to make a confession: When it comes to dealing with numbers, I'm a real grunt. My brain just can't handle 'em. Faced with columns of numbers, my mind shifts into dyslexia—which makes trying to keep my checking account straight a real trick.

I am further embarrassed to confess that, until recently, I have kept my Atari 8-bit system set up because its Financial Wizard program has balanced my accounts for the past three years! It seems the time is right to see what WordPerfect's math and macros can do.

Checkbook Balancer is the result of my initial experimentation with WordPerfect's advanced features. I wouldn't call it a work of art, but it does the job quite capably. Checkbook Balancer is a document and set of associated macros which allow you to enter and store new checks and deposits, keep separate lists of cleared and outstanding transactions, and do all the (Ack! Phht!) math to balance your books. In addition, its macros allow you to print a hard-copy list of all outstanding items, or just the dated summary statement. Not too shabby, for a word processor.

SETTING UP

The first thing to do is prepare a place for the Checkbook document and its associated macro files. You may want to set aside a separate disk, or simply create a directory on an existing disk. To keep things simple, name the disk or directory "CHECKS."

Once you have done this, set the default directory to the newly created area. Select List Files (F5) and press Return at the first requester. When the directory listing appears, press 7 to change directories. If you have

created a separate disk to store the Checkbook document, enter Checks: and press Return. If you created a directory, enter the appropriate path.

We're setting aside a location "away" from other word processing files because we will be making use of handy AMIGA key macros. WordPerfect stores user-defined macros on disk, and while you may define a virtually unlimited number of macros with names two characters or longer, the more convenient AMIGA key macros are limited to 26 per directory.

If you haven't configured your version of WordPerfect to accept the maximum number of AMIGA key macros, you should do that now. The programmers who translated WordPerfect to the Amiga had a problem: WordPerfect normally uses the ALT key to call one keystroke-named macros (e.g., ALT-A, ALT-Q, ALT-Y). In the Amiga's case, the ALT key was reserved for use in conjunction with foreign character sets, so the programmers opted to use the AMIGA keys to call macros. The AMIGA keys are the red "A" keys on either side of the space bar on the Amiga 1000 and 2000; the left AMIGA key on the Amiga 500 is marked with a Commodore logo.

Amiga WordPerfect allows you to decide which keys are available to the Amiga, and which can be used to name macros. To assign macro keys, call up the Screen menu (CTRL-F3) and select 2, Ctrl/Amiga keys. This

(continued)

calls up a large requester with four columns of key combinations. The right two columns set the AMIGA key assignments; as configured out of the box, some of these key assignments read AMIGA and some read MACRO. To make an AMIGA key combination available to WordPerfect, click on the letter. The requester box next to the letter turns orange; enter a "1" and press RETURN. The key assignment should now read MACRO. For this article, you must change the AMIGA-C and AMIGA-T to MACRO; you may change others as well if you like.

Unlike many "hotkey" combinations common to Amiga programs, WordPerfect does not usually differentiate between the left and right AMIGA keys. That is, a macro defined using the right AMIGA key may be invoked using the left AMIGA key and vice versa. The AMIGA-M and AMIGA-N keys are the odd exception to this rule. The Amiga's operating system uses left AMIGA-N and left AMIGA-M to switch between screens, when more than one screen is open. There is nothing WordPerfect's programmers can do about this—the operating system intercepts these key combinations before WordPerfect "sees" them. If you want to use AMIGA-N or AMIGA-M as macro names, remember to define and invoke them in conjunction with the right AMIGA key.

Once you have set the AMIGA key assignments, click on the Accept gadget to continue.

STEP BY STEP

We are now ready to begin building the Checkbook document. Because the chances for error are great, we'll approach document construction step-by-step. Follow these instructions carefully, and remember, computers only look smart. In reality, they're pretty dumb and extremely literal. You may want to refer to Figure 2 from time to time; that's how the document will look when we're done.

In this article, I will reference WordPerfect functions by their keystrokes; however, most of the functions have mouse-and-menu equivalents you may use if you prefer to point-and-click. The menu selections may be used to build both the document and the macros.

One more ground rule before we begin. Throughout this article I will denote words and phrases to be typed by boldfacing them—for instance, "Type **Page heading.**" The boldfacing is there for clarity's sake—do not boldface the words when you enter them.

BUILDING THE DOCUMENT

1. First, let's set some margins. Call up Line Format by pressing SHIFT-F8, then press 3 to select Margins. Set the left margin to 0, and the right margin to 78.

2. Next, set tab stops to keep everything aligned. Press SHIFT-F8, then 1. When the Tab Set requester appears, press CTRL-END to clear all tab stops. Then, using the cursor keys, position the cursor under tab position 9 and press T. Repeat this process to create tab stops at positions 26, 40, and 46, then click on Accept to continue.

3. In order for WordPerfect to calculate columns of numbers, the columns must be defined. Call Math/Columns by pressing ALT-F7, then select option 2, Math Def.

The Math Definition menu allows you to tell WordPerfect what kind of information you will store in each column. An exception is the first column, which is always treated as text. The column heading "A" refers to the second column, "B" to the third, and so on.

The first column in our document will contain the check number, which will not be calculated; so column 1 is well suited for a text column. Column 2,

which will contain the date, is also a text column. To define the second column, move the cursor under "A" on the Type line and press Return. Select 2 to set column A to text.

Next, use the cursor keys to move to column B. As this column will contain the amounts of any checks, it will be a numeric column. The 3 under the B signifies a numeric column, so no change is necessary.

The fourth column, denoted by the C, will contain deposit amounts. This will be a numeric field, so the 3 may stay as it is.

We will use the final column for remarks such as the check payee. This will be a text column, so move the cursor under the D and press Return. Select 2 (Text), then click on Accept.

4. The Math/Columns requester should still be on screen at this time. (If you jumped the gun and clicked it off, recall it now using ALT-F7.) Select 1 to turn the math function on. The word "Math" appears at the bottom of the screen.

5. Check your work. Press ALT-F3 to examine the formatting codes you just entered. The Reveal Codes window should contain something that looks like this:

```
[Margin Set:0,78][Tab  
Set:9,26,40,46][Math Def][Math On]
```

Press the space bar to exit Reveal Codes.

6. The next thing you must do is enter the section and column headings for the outstanding transactions, as shown in Figure 1. After all these keystrokes, we will finally do some text formatting! The first operation is pretty simple: We will center the title at the top line of the document.

To turn centering on, press SHIFT-F6. The cursor should jump to the center of the line. Next, type the words *** Outstanding Transactions ***. (Note that WordPerfect centers as you type.) When you have typed this heading, press Return twice to turn off centering and move down two lines in the document.

Formatting is critical for the macros to work correctly; be careful to observe upper and lower case, and the spaces between the asterisks and the heading. WordPerfect doesn't particularly care how you type things, but our macros will be using the Search function to navigate the document. Enter the column headings shown in Figure 1 (Check#, Date, etc.), using the space bar to align the headings in the correct columns. You can't use the Tab key in this case because the tab stops that correspond to numeric columns will always align on a decimal point (period), and so will not properly align the headings.

7. Next, type in the five test data transactions shown in Figure 2. (These are the three checks and two deposits entered in the Outstanding Transactions section.) When entering transactions, you must use the Tab key to separate entries on a line. WordPerfect uses tab-aligned columns to keep track of what to calculate, and using the space bar to separate fields will result in unpredictable results when attempting math functions. To leave a field blank, press Tab to move past it.

Once you have entered a transaction (one line of data, either a check or a deposit), using Tab to move the cursor on that line will alter the formatting. Instead, press the CTRL key in conjunction with the left and right cursor keys to move the cursor left or right one word at a time.

8. After typing the word **Paycheck** on the last line of test data, press Return. Enter a line of dashes from position 0 to position 77, under the column headings. Press Return again.

9. Now, type *** TOTALS *** (all capital letters this time, and watch those asterisks!). Press Tab once; this will position the cursor under the Check Amt column. Enter a + (plus sign), then Tab over to the Deposit Amt column. Enter another + and press Return. The plus signs tell WordPerfect where to display the results of summing a column.

10. Once you have entered the two plus signs, call the Math/Columns option by pressing ALT-F7. Click on Math Off. Math Off signals the end of a math calculation area; to keep everything straight, we will be defining separate calculation areas for outstanding transactions, cleared transactions, and a summary report.

11. After you have entered the Math Off, press Return twice, press SHIFT-F6 to activate centering and enter, *** Last Outstanding Transaction ***.

12. To complete the Outstanding Transactions section, type CTRL-Return to enter a forced, or "hard," page break.

CLEARED TRANSACTIONS

13. We will now construct the section that will be used to hold cleared transactions. Press ALT-F7, then click on Math On to begin the new calculation area.

14. Press SHIFT-F6 to turn centering on, type *** Cleared Transactions ***, and press return twice.

15. Enter the same headings (Check#, Date, etc.) you entered for the Outstanding Transactions section. Remember to use the space bar, not the Tab key, to align the headings.

16. Enter a Return at the end of the line of dashes under the column headings. On the line immediately under the dashes, enter a solid line of dashes from the left margin to column 77. When you have done this, press Return.

17. Repeat steps 9 and 10 to enter the line for the totals, and turn Math Off. Press Return twice.

18. Press SHIFT-F6 to begin centering, then type *** Last Cleared Transaction ***. End this section with a hard page break (CTRL-Return).

THE SUMMARY PAGE

19. We will now build the final section, which will display the check-book balancing summary. Its format will be different from the preceding two sections.

We will begin by resetting the tab stops; press SHIFT-F8, then 1. Press CTRL-END to clear the existing tab stops, and enter new tab stops at positions 9 and 51. Click on the Accept gadget to continue.

20. Press ALT-F7, then 1 to turn Math On.

21. Press SHIFT-F6 to begin centering, then type *** Summary ***. Press Return twice.

22. Hit Tab, and enter the word **Date:**. Press Return twice.

23. Hit Tab again, then type **Balance Shown on Statement:**. Press Return.

24. Hit Tab and type **Outstanding Deposits**. Using the space bar to enter pad spaces (do not use the Tab key), align a : (colon) under the colon on the line above. Press Return. Follow the same procedure for the **Less Outstanding Checks** label. When you have entered its colon, press Return.

25. Press Tab, then enter a line of dashes from position 9 to position 55. Press Return.

26. Press Tab, then type **Adjusted Statement Balance:**. Press Tab, then enter a plus sign. Press Return. Finally, press ALT-F7, then 1 to turn Math Off.

Congratulations—the Checkbook document is done! Now the real fun starts...

MACRO MAGIC

Mention the word “macro” and a lot of people break out in cold sweats, with visions of the cryptic, code-laden commands used by programs such as Lotus 1–2–3. If you are among the ranks of the techno-terrified, you can relax. WordPerfect macros are easy to construct and even easier to use.

The word “macro” simply means “big.” In “computerese,” it refers to a short instruction or command that takes the place of many commands. If you ever typed “Dir” from the Amiga Command Line Interface (CLI), you’ve executed a macro. If you ever programmed a line as simple as PRINT “HELLO!” in Basic, you’ve used a macro. Entering Dir from the CLI or PRINT from Basic causes those one-word commands to set in motion a multitude of smaller commands which execute the task.

While many people have used macros, few have actually constructed them. In WordPerfect, a macro simply represents a series of keystrokes entered in sequence. The keystrokes may consist of text characters or WordPerfect commands. When the macro is executed, WordPerfect “plays back” the keystrokes in the same sequence they were entered when the macro was defined.

It’s a lot easier to build a macro than it is to explain the theory, so let’s give it a try.

THE NEW MACRO

Our first macro will be one you can use with the Checkbook document. Its function is simple: It creates a blank line at the end of the Outstanding Transactions list to allow you to enter a new check or deposit. You could do essentially the same thing by positioning the cursor at the line of dashes at the end of the Outstanding Transac-

tions list and hitting the Return key, but our macro will be a little smarter than that.

What would happen, for instance, if you mistakenly added a new transaction to the end of the Cleared Transactions list? Or pressed Return in the middle of a line? Either case could cause serious problems, so the macro we are about to build will be smart enough to keep that from happening. As a matter of fact, even if you were to invoke the macro from the middle of the Summary page, the macro would search out the end of the Outstanding Transactions list, open up a blank line for the new transaction, and position the cursor at the beginning of the line, ready to go. That’s service!

DEFINING THE NEW MACRO

If the Checkbook document is not on-screen, retrieve it at this time. To check the “smarts” of our new macro, position the cursor anywhere on the Summary page.

Press CTRL-F10 to initiate Macro Define. A requester labelled Define Macro appears in the center of the screen with the instruction, “Enter macro name.” Type the word NEW in the requester, and press Return. The requester disappears. At the bottom of the screen, the words “Macro Def” appear.

Since there is no way to assure that the person executing this macro will do so from the correct place in the document, we need to reset the cursor to a predictable, “neutral” position, then work from there. The top of the document is ideal for this. To get there, press the HOME key twice, and then the up-arrow key.

We will use WordPerfect’s Search function to locate the end of the Outstanding Transactions list. WordPerfect can search in either the forward or reverse direction. We want a forward search, so press F2.

When prompted to enter a search string, type *TOTAL. When you have entered the search criteria, you may either click on the Accept gadget or press F2 to begin the search. Do not press Return to end the search string! If you do, WordPerfect will try to find a string that ends in a Return!

As a result of the search, the cursor should now be on the “S” in *TOTALS *. The next thing we want to do is move the cursor up one line, so press the up-arrow cursor key. Next, the cursor must be positioned at the beginning of that line, so press Home twice, then press the left-arrow cursor key.

To open up a blank line, press Return. Finally, press the up-arrow cursor key to position the cursor on the blank line. That is the last instruction of the macro, so press CTRL-F10 to end macro definition. That’s it!

Since we don’t really want a blank line at the end of the Outstanding Transactions list, press Del once to close the line. We are now ready to test the macro.

TESTING THE NEW MACRO

First, place the cursor anywhere in the document, or even at the correct place at the end of the Outstanding Transactions, if you wish. Press ALT-F10, Macro. A requester opens, asking you to enter the name of the macro. Type NEW and press Return.

WordPerfect will pause; then, if you’ve entered everything correctly, it will open a line at the bottom of the Outstanding Transactions list and place the cursor at the first position of the new line. Press Del to close the line.

WordPerfect stores its macros on disk, in the current default directory, which means once you put together a macro you like, you need not enter it again. Multi-letter named macros like NEW are stored with the macro name and a

.MAC extender; if you select List Files now you can see NEW.MAC in the Checks directory. Key-combination macros are given more cryptic file-names; for example, a macro invoked by pressing AMIGA-T appears as {WPL}___.AMGT.

MACRO ENTRY

You are now ready to enter the remainder of the macros, listed in Tables 2 through 5. The sidebar accompanying this article, *Entering The Checkbook Macros*, should save you some problems. In addition, there are some unique considerations to each macro.

THE CLEAR MACRO

This macro, listed in Table 2, transfers transactions from the Outstanding to the Cleared section of the document, then recalculates the totals for both sections. It must be called from a line containing an outstanding transaction or great problems will arise! There's really no way around this, because the whole point of the macro is presumably to "clear" a specific item.

This macro and the Total macro which follows are both AMIGA-key macros; that is, to execute them once they're defined, you need only press AMIGA-C to call the Clear macro, or AMIGA-T to call the Total macro. The names are entered differently than was done with the New macro; follow the table listings and you won't have any problems.

Before entering this macro, position the cursor on the line containing the "Computer Experience" transaction. When you have completed entering this macro, this line should be transferred to the Cleared section, with no blank line in the Cleared section. In the process of defining this macro, WordPerfect will calculate new totals for the Outstanding and Cleared

sections. If you have done everything right, your totals should add up like this:

Outstanding checks:	35.00
Outstanding deposits:	1,345.00
Cleared checks:	115.00
Cleared deposits:	0.00

THE TOTAL MACRO

This macro calculates totals in all sections, including the summary page, where it enters the current system date. It may be invoked from anywhere in the document, and it doesn't matter where the cursor is when you begin defining the macro.

At the conclusion of the Total macro, the figures at the end of the Outstanding and Cleared sections will not be changed from those listed above. In addition, the Summary section will contain the following values:

Balance Shown on Statement:	
(blank-no value)	
Outstanding Deposits:	1,345.00
Less Outstanding Checks:	-35.00
Adjusted Statement Balance:	1,310.00

In actual operation, you would enter the closing balance shown on your bank statement on the "Balance Shown" line before executing this macro.

THE PRINT-OUT MACRO

This macro, listed in Table 4, provides a hard-copy list of all outstanding checks and deposits, plus total figures for each. It may be defined and invoked from anywhere in the document.

There's one minor glitch to this macro and the one that follows: Both macros call WordPerfect's PRINT program and leave the Printer Control display on screen when they exit. I have not yet discovered a graceful way to get the macro to "click off" this screen, so for the time being you must do that manually. If anyone can figure out a way around this, I'd like to hear it.

THE PRINT-SUM MACRO

This very simple macro, listed in Table 5, recalculates the totals for the Outstanding and Cleared transactions, copies the appropriate figures into the Summary page, calculates the Summary page, and sends it to the printer. Yet it does all this using only three instructions!

The secret is, one of the instructions is itself a macro—AMIGA-T, the TOTAL macro. The PRINT-SUM macro first calls the TOTAL macro, which recalculates all sections in the document. The next instruction ensures the cursor is positioned on the last page of the document (this will always be the Summary page), then prints that page.

Do not be concerned if this macro does not appear to work when you initially type it in. The AMIGA-T macro will not execute during definition, but it will work correctly when invoked.

BALANCING THE BOOKS

With the Checkbook document and all macros complete, you should test the macros to verify they are working correctly. When you are satisfied with your handiwork, you can put the Checkbook document to use!

The first thing to do is delete the dummy data we used to test the system. Be sure you do not delete the plus signs (+) on each TOTAL line—WordPerfect needs these to know where to sum the columns.

Next, you will have the unenviable task of entering your checks and deposits into the Checkbook document. Be sure to place the cleared and outstanding items in the correct sections.

When you enter new checks and deposits, it is important that you use the Tab key (not the space bar) to move from one column to the next.

WordPerfect's math functions require that all figures to be calculated be arranged in columns delimited by tab stops. If the totals in your document are flaky, this is the first thing you should investigate. You can quickly search for tabs by examining the document under Reveal Codes (ALT-F3).

To reconcile your checkbook against your bank statement, first enter the ending balance (as shown in the statement) on the Summary page in the column corresponding to Balance Shown on Statement:. Use the Tab key to align this entry at the first tab stop after the colon.

Next, clear any transactions—checks and deposits—which appear on the bank statement. To clear a transaction, simply position the cursor anywhere on the line containing the transaction and press AMIGA-C.

When all the appropriate items have been cleared, you are ready to calculate the final totals. To do so, press AMIGA-T. This macro performs the actual checkbook balancing operation. If the Adjusted Statement Balance on the Summary page matches your checkbook balance, congratulations! If not, there's a mistake somewhere—perhaps the bank's error, perhaps an entry error, maybe a typo in the Checkbook document. At any rate, you're probably in for some hard time slaving over a hot calculator (Ack!).

If all goes well, you may generate a list of all outstanding transactions by pressing ALT-F10 to invoke the macro, then typing PO and Return when prompted for a name. If all you want is a hardcopy listing of the Summary page, enter PS when asked for the macro name.

ONLY THE BEGINNING

While Checkbook Balancer is an impressive demonstration of WordPerfect's math and macro capability, the Checkbook document itself

is pretty basic, and there is much room for enhancements. For instance, how about creating a macro to print a list of tax-deductible expenses, keyed to a /T/ string in the Memo field?

If you found this article useful, I would appreciate any suggestions on future WordPerfect tutorials. If you got stuck at some point, I'll be happy to help you get unstuck. In any case, I can be reached at the E-mail addresses listed at the head of this article. May you encounter good luck teaching WordPerfect its balancing act!

•AC•

The author would like to thank Mark Hamilton and Lynn LaBaron of WordPerfect Corporation for their assistance and technical review in preparing this article.

* Outstanding Transactions *									
Check#	Date		Check Amt		Deposit Amt		Memo		
^	^		^	^	^	^	^	^	^
5	9		16	20	28	32	42	46	77

Figure 1

Checkbook Document Heading Numbers and arrows denote column positions; do not type them in.

* Outstanding Transactions *				
Check#	Date	Check Amt	Deposit Amt	Memo
0000	01/01/88		345.00	Balance carried forward
0001	01/01/88	10.00		Happy Jack's Used Disks
0002	01/02/88	115.00		The Computer Experience
0003	01/02/88	25.00		StarShip Amiga
	01/10/88		1,000.00	Paycheck
* TOTALS *		+	+	
* Last Outstanding Transaction *				
* Cleared Transactions *				
Check#	Date	Check Amt	Deposit Amt	Memo
* TOTALS *		+	+	
* Last Cleared Transaction *				
* Summary *				
Date:				
Balance Shown on Statement		:		
Outstanding Deposits		:		
Less Outstanding Checks		:		
Adjusted Statement Balance		:	+	

Figure 2 The Checkbook Document

ENTERING THE CHECKBOOK MACROS

In the interests of lowering the frustration levels of Amiga users everywhere (and precluding an excessive number of anguished phone calls to WordPerfect technical support), I have followed what I consider to be the clearest possible format for presenting macros, seen in Tables 1 through 5. To enter a macro, simply enter the commands and keystrokes in the order listed.

Specific conventions are observed throughout the tables. For the most part they're pretty obvious, but some could cause confusion to the beginner.

WHEN THE LISTING SAYS... IT MEANS:

UP	Up-cursor key
DOWN	Down-cursor key
->	Right-cursor key
<-	Left-cursor key

Type text string

Type the text displayed in bold face; do not boldface the text as you type it. Upper and lower-case DOES matter, as do spaces.

FUNCTION (Four times)

Execute the FUNCTION the number of times specified.

(comment)

Anything in parenthesis are informational remarks and shouldn't be typed.

SHIFT-F8

Press and hold the SHIFT key (or ALT, CTRL or AMIGA key) and press the function key given.

CTRL-F4,2

Execute the first part of the instruction, then select the option number specified when the requester appears. In this example, pressing CTRL-F4 calls the Move Text menu; when the menu appears, select opt. 2, Copy.

MORE MACRO TIPS

Until computers reach the point where they're smart enough to do what we mean, and not what we tell them to do, bugs will be with us. Because WP macros act on the document as you define them, be sure to save the file before testing a macro. Here are more tips that will save you problems.

Mistakes: WordPerfect's Macro Define function records everything—even mistakes. If, for instance, you want the macro to type the word Hello and mistakenly type Hellp, you can backspace over the "p" and replace it with an "o," but the edits will be recorded in the macro along with the "good" stuff. In the case of the Hellp typo, there would be no affect on macro operation, but the extra instructions cause the macro to execute longer.

In addition, there is currently no way to edit a macro composed with Amiga WordPerfect. The MS-DOS WP users have had a separate macro editor available for a while now, but at this time WP Corp has no plans to develop one for the Amiga. It's no big deal for simpler macros, but after you retype the TOTAL macro a few of times to fix bugs, you'll see why a macro editor would be handy. Let's hope WP Corp changes its mind on this one.

The bottom line is, if you make a mistake during macro definition, all you can do is bail out and start over. To exit Macro Define, press CTRL-F10.

Search Within Macros: The way WordPerfect defines macros allows the unsuspecting user to build at least one pernicious trap into the macros—a trap that may not even manifest ITSELF until a later word processing session.

Here's how it works. Let's say your macro needs to search out the word Aardvark as part of its routine. You press F2 to begin the search, type Aardvark in the search requester, and press F2 again to execute the search.

A few keystrokes later in the macro definition, you mistype a command and decide to start over. When you get to the search portion the second time around, you press F2 and the requester appears with Aardvark still displayed as the search criteria. "Yeah buddy," you think, pressing F2 again to execute the search. The macro definition continues, this time to a successful conclusion. You even test the macro, and it works fine.

Two days later when you invite a friend over to impress him with your wizardry, you invoke the macro and it doesn't work at all—as a matter of fact, instead of searching on Aardvark, it stubbornly searches on a completely different word. What's going on?

Well, WordPerfect is simply doing exactly what you told it to do. The first time the macro was defined, WordPerfect recorded:

(Beginning of macro) F2 Aardvark F2
(End of macro)

However, the second time the macro was defined, WordPerfect recorded:

(Beginning of macro) F2 F2 (End of macro)

In other words, Macro Define did not actually record any search criteria; all it knew was, F2 was pressed twice in a row, so that's what it recorded.

It worked when the macro was composed because, in the absence of new search criteria, WP will always search on the last criteria given.

Nasty? You bet. Just another instance of them darn computers doing exactly what they're told. But it's easy to avoid—remember, when defining a macro, ALWAYS type out the search criteria, even if it means retyping exactly what appears in the requester.

- Steve Hull

TABLE 1

Macro name: NEW

Function: Creates a blank line at the end of the Outstanding Transactions list.

To invoke: Press ALT-F10, then type NEW; may be invoked from anywhere in the document.

Macro key sequence	Explanation (Do not type)
CTRL-F10	Begins Macro Define
Type NEW	Enter macro name
RETURN	Signify end of macro name
HOME HOME UP	Cursor to top of document
F2	Forward search
Type * TOTALS	Enter search criteria
F2	Execute search
UP	Cursor up one line
HOME HOME <-	Cursor to beginning of line
RETURN	Make a blank line
UP	Place cursor at new line
CTRL-F10	End Macro Define

TABLE 2

Macro name: CLEAR

Function: Transfers one transaction from Outstanding to Cleared section; returns cursor to the next entry in the Outstanding list.

To invoke: Press AMIGA-C. Must be invoked from the line containing the transaction to clear; cursor may be at any position on that line.

Macro key sequence	Explanation (Do not type)
CTRL-F10	Begin Macro Define
AMIGA-C	Assign macro name
HOME HOME <-	Cursor to beginning of line
ALT-F4	Block on
END	Cursor to end of line
CTRL-F4,1	Cut marked text
Type @@@	Enter place keeping marker
F2	Forward search
Type * Cleared	Enter search criteria
F2	Execute search
F2	Forward search
Type * TOTALS	Enter search criteria
F2	Execute search
UP	Move cursor up one line
HOME HOME <-	Cursor to beginning of line
RETURN	Make room for new item
UP	Move cursor to new space
CTRL-4,5	Retrieve stored transaction
ALT-F7,2	Calculate new Cleared total
SHIFT-F2	Reverse search
Type @@@	Enter search criteria
F2	Execute search
BACKSPACE (4 times)	Delete place-keeping marker, close line
HOME HOME <-	Cursor to beginning of line
DOWN	Position cursor on next item
ALT-F7,2	Calculate new Outstanding total
CTRL-F10	End Macro Define



Money Mentor™ has a New Engine

Climb Aboard the new "C" version of Money Mentor™ for the ride of your life. Speed is your ticket to faster data input and dazzling graphics output. If your destination is better control of your personal finances, there's no faster way to get there than with Money Mentor™.

A unique system called "Smart Scrolls" handles a diversity of tedious data entry functions and can save 70% of the typing typically required for entry.

Money Mentor™ features:

- Net Worth Statement.
- 200 budget categories.
- 30 integrated accounts: checking, cash, saving and credit cards.
- Elaborate search routine allows editing of transactions according to your specific guidelines.
- Automatic check printing.
- Automatic Account Balancing.
- Colorful graphic reports illustrating actual versus budgeted amounts.
- Over 50 reports from which to choose.

Let Money Mentor™ put your finances on the right track... FAST!



11844 Rancho Benardo Rd; Ste.#20
San Diego, CA 92128



To order,
call (619) 451-0151



TABLE 3

Macro name: TOTAL
Function: Calculates totals in Outstanding, Cleared, and Summary sections; stamps Summary page with current date.
To invoke: Press AMIGA-T. May be invoked from anywhere in the document.

Macro key sequence	Explanation (Do not type)
CTRL-F10	Begin Macro Define
AMIGA-T	Assign macro name
HOME HOME UP	Cursor to top of document
F2	Forward search
ALT-F7,2	Search for MATH ON code
F2	Execute search
ALT-F7,2	Calculate Outstanding totals
F2	Forward search
Type * TOTALS	Enter search criteria
F2	Execute search
CTRL-RT ARROW (twice)	Move cursor to total for outstanding checks
ALT-F4	Block ON
F2	Forward search
Type . (a period)	Enter search criteria
F2	Execute search
-> (twice)	Block cents digits
CTRL-F4,2	Copy marked text
F2	Forward search
Type Outstanding Checks	Enter search criteria
F2	Execute search
F2	Forward search
Type : (a colon)	Enter search criteria
F2	Execute search
CTRL-END	Delete to end of line
CTRL-F4,5	Retrieve outstanding dept ttl
TAB	Place total in numeric column
Type - (a dash)	Denote figure as a negative number search
Type * Outstanding	Enter search criteria
F2	Execute search
F2	Forward search
Type * TOTALS	Enter search criteria
F2	Execute search
CTRL-RT ARROW (3X)	Move to Outstanding Depts ttl
ALT-F4	Block ON
F2	Forward search
Type . (a period)	Enter search criteria
F2	Execute search
-> (twice)	Block cents digits
CTRL-F4,2	Copy marked text
F2	Forward search
Type Outstanding Deposits	Enter search criteria
F2	Execute search
F2	Forward search
Type : (a colon)	Enter search criteria
F2	Execute search
CTRL-END	Delete to end of line
CTRL-F4,5	Retrieve outstanding dept ttl
TAB	Place total in numeric column
SHIFT-F2	Reverse search
Type * Cleared	Enter Search criteria
F2	Execute search
ALT-F7,2	Calc. ttls for cleared items
F2	Forward search
Type Date:	Enter search criteria
F2	Execute search
CTRL-END	Delete previous date entry
(Press sp. bar once)	Allow room after colon
SHIFT-F5,1	Insert date
ALT-F7,2	Calculate totals
CTRL-F10	End Macro Define

TABLE 4

Macro name: PRINT-OUT
Function: Recalculates totals for the Outstanding Transactions section and sends a list of all outstanding transactions to the printer.
To invoke: Press ALT-F10, then type PO. May be invoked from anywhere in the document.

Macro key sequence	Explanation (Do not type)
CTRL-F10	Begin Macro Define
Type PO	Enter macro name
RETURN	Signify end of macro name
HOME HOME UP	Cursor to top of document
F2	Forward search
ALT-F7,2	Search for MATH ON code
F2	Execute search
ALT-F7,2	Calculate totals for the outstanding transactions
ALT-F4	Turn Block on
F2	Forward search
Type Transaction *	Enter search criteria
F2	Execute search
SHIFT-F7,Y	Print block
HOME HOME UP	Return to top of document
CTRL-F10	End Macro Define

TABLE 5

Macro name: PRINT-SUM
Function: Recalculates all sections, updates Summary page and sends it to the printer. Requires the TOTAL macro (Table 3) to run.
To invoke: Press ALT-F10, then type PS. May be invoked from any point in the document.

Macro key sequence	Explanation (Do not type)
CTRL-F10	Begin Macro Define
Type PS	Enter macro name
RETURN	Signify end of macro name
AMIGA-T	Invoke TOTALS macro
HOME HOME DOWN	Cursor to end of the document
SHIFT-F7,2	Print page
CTRL-F10	End Macro Define

FORMS IN FLIGHT

A 3D rendering and animation package for the Amiga™

by Stephen R. Pietrowicz

Micro Magic's Forms In Flight (FIF) is one of the 3D rendering and animation packages now available for the Amiga. It was released several months ago and has already gone through several updates. FIF renders objects in either wire frame (only the outlines of the objects show) or solid objects. Solid objects can be shaded, and you can view objects in any direction by rotating them. Once objects are rendered, you can set up animation sequences to animate them on screen. FIF has some very nice features, but there are still several problems with the package.

According to Micro Magic, the program requires one megabyte of memory; that's the absolute minimum. Once FIF is fully loaded, it leaves only about 160K of memory free in a one meg system. If your system has only one meg, I suggest you run *no* other programs in order to leave enough room for pictures to display in the program. I'll explain this more when I talk about playing the animations.

Reading The Manual

The manual comes in a three-ring binder you can lay flat your desk as you learn to use the program. The manual is split into three sections: Getting Started, Commands, and Animation. It also has an appendix explaining the different input modes of the program.

Getting Started includes a brief tutorial on how to create and rotate a regular 6-sided polygon. It also explains how to change the default colors and remove the hidden lines in the wire

frame objects. The manual is very simple and leads the reader through the first example as if this were the first time the reader had used an Amiga. I thought the first example was quite good; it didn't presume anything about the user.

Unfortunately, the rest of the manual uses this approach as well. For nearly every command, you are presented with all the information you need to learn about that command without having previous knowledge about how to use any other command. Sections of text are frequently repeated. That's good if you're looking up individual commands for reference, but it makes it very difficult to read through the manual to learn how to use the program.

Many examples require that certain values be input from the keyboard (or with the mouse), and they carry the instructions a bit too far. One example in the manual states, "Set up the step size to 800 by clicking the left button on the '8,' the '0,' the '0,' the 'USE' box, and then the 'OK' box."

This is appropriate in an introductory section describing the different input modes of the program, but is not necessary for every on command requiring numerical input.

Since the commands are presented in alphabetical order—rather than in a more meaningful order—it is difficult to learn the program. The manual would have been much better if it presented the information in a series of examples, starting with a simple example, and proceeding to more and more complex examples.

The Animation section of the manual describes how to animate a scene with a merry-go-round. This section does a good job of explaining how to animate the objects, and the example is a valuable tool to learn how to animate your own scenes. It would have been nice to see more than one example animation. I really wish the rest of the manual were presented in this fashion.

The manual's appendix explains the input modes. This section would have been much more appropriate in the beginning of the manual; the reader wouldn't be forced to read through all the over-simplified examples.

Creating Scenes

When the program starts, it comes up in high resolution interlace mode, with dark blue letters on a black screen.

Creating scenes isn't very difficult, but can be a bit confusing until you learn to think in terms of X/Y/Z planes. With practice, you can create interesting looking objects.

When creating objects in 2 dimensions, you'll probably draw free-hand polygons most often. A crosshair appears on the screen to let you draw the outline of your object. Once the object is completed, you can draw another. (Objects can also be rendered in 3 dimensions.) Objects can be rendered in the XY, XZ, YZ plane, or in a user defined plane (at other angles besides XY, XZ, and YZ).

(continued)

If you'd like to draw regular polygons (triangles, squares, pentagons, etc.), a command to create those objects is available just by typing the number of sides you wish draw. Spline curves can also be drawn.

Once polygons and objects have been rendered, you can select entire groups of objects or individual objects in a scene and arrange them to obtain the object formation you want. On-screen objects can be rotated and observed from any direction. Scenes you create can be saved to disk and retrieved for later use, allowing you to create a library of objects.

An interesting feature in FIF is the ability to create red/blue stereo images. It's a very nice touch. Using this feature, you can create interesting 3D effects to be viewed with red/blue 3D glasses. The glasses aren't included with the program, but you can get them in many toy stores.

According to the manual, FIF can support the HP-GL plotter. I wasn't able to test this feature because I don't own a plotter.

One thing I don't like about the program's user interface is that it doesn't keep you in the mode you selected. For example, I wanted to select several different objects on the screen using the SELECT BOX AROUND function. Instead of leaving the program in a mode where I could select one object and then select another while still in the SELECT BOX AROUND mode, I was forced to return to the menu to turn this mode on again. It would be much more useful to leave the program in a selection mode until the user is done selecting objects, and then allow him to exit that mode.

Creating Animations

After you build a scene, you can proceed to the animation description screen. Interaction and relationships between objects in the scene are

described in a graphic "tree" display. Objects that depend on each other are connected by "branches."

For example, imagine that an arm with a hand is rendered on the screen. If the forearm is moved in any direction, the hand is automatically redrawn in the correct position. This touch makes it much easier to move entire scenes without worrying about moving each individual object.

Motion of objects or groups of objects is defined in terms of how far they move forward, down, left, or rotate down, left, or roll. To move in the opposite direction, you must give the negative offsets of the appropriate commands. For example, to rotate left ten degrees, you specify "10" with the rotate left command; to rotate right ten degrees, you specify "-10" with the rotate left command. "Rotate roll" isn't explained in the manual and is left for the user to figure out.

When scenes for animations are created, they are stored on disk in separate files. An animation with 70 frames will have 70 separate files. FIF does not have any way to load or save pictures in IFF format, although Micro Magic notes they may support IFF backdrops in the future.

Playing Animations

Once an animation is created, you can either playback the animation from within FIF, or you can use FastFlight, a program that comes with FIF. Each has its problems.

If you run animations from within FIF, the frames are loaded individually from disk, and playback is very jerky. FIF tries to solve the jerky display problem with the separate FastFlight program. FastFlight loads the animation frame-by-frame and displays each (as in FIF itself). Once all the frames are in memory, the animation can be shown at a relatively good speed.

A playback mode that works at the speed FastFlight would be useful so the user isn't forced to toggle back and forth between the two programs. Even if your system does have more than one meg, it isn't practical to run both FIF and FastFlight at the same time. FastFlight's instructions say you should reboot your system before and after each animation is run.

I asked a Micro Magic representative why FIF doesn't have the playback mode, and why FastFlight must be used. He stated that because FIF uses so much memory, FastFlight couldn't be included in the program. Instead of trying to add that capability, they decided to make it a separate program to keep the memory requirements of the program down to one meg.

Micro Magic also stated that people with systems with more than one meg can run FIF and FastFlight together. FastFlight has a bug, however: The program does not free all the memory. The manual's rebooting suggestion ensures that the machine won't crash because it has run out of memory. If you have enough memory to run FIF and FastFlight together, Micro Magic suggests you save your animations frequently, so you don't lose data if the machine crashes. At this time, Micro Magic does not have plans to create a version of FIF with FastFlight capabilities for users with machines with more than one meg, but they are currently looking into the FastFlight bug.

Overview

All in all, FastFlight's manual needs some more work, and some of the user interfaces could be better. Micro Magic told me they plan to come out with a better manual and another update for current owners of Forms in Flight. Upgrades will then be made available for a modest fee. With a better manual and the promised improved versions of the program, I think Micro Magic is getting on the right track.

•AC•

Silicon Dreams and Jewels of Darkness: A Touch of Nostalgia

reviewed by Kenneth E. Schaefer

Long ago—before IBM figured out that computers could sit on desks and mice could learn to paint, back when you could express a personal computer's memory in a two-digit number and the number of colors it displayed in one digit—the first graphic text adventures appeared. Games like *Wizard and the Princess* and *Blade of Black Poole*, from companies like Adventure International and Sierra On-line, had very simple parsers, and the graphics were static images, usually in the upper two-thirds of the screen. By today's standards, both the parser and the graphics in those games were below par. But this was several years ago, and as a wise man once said, "The only thing constant about the computer industry is change."

Now Firebird Software (that's Rainbird in British, I guess) has brought back the classic graphic text adventures, with the introduction of two products for the Amiga: *Silicon Dreams* and *Jewels of Darkness*. *Silicon Dreams* is a trilogy of science fiction adventures set around and on the world of Eden, while *Jewels of Darkness* is a trilogy of fantasy adventures set on a swords-and-sorcery world called Valaii. The adventures are related in at least a tenuous way, but they all stand alone as separate games that can be played in any order. I must admit, however, that I liked them better when I played them in order. It seemed more logical that way. There is also the possibility of achieving the title "Supreme Adventurer" if you carry your score from one game to the next.

The *Silicon Dreams* games follow the exploits of Kim Kimberly. In "Snowball," the first game, you (as Kim) are awakened early from your frozen slumber. You are aboard Snowball 9, a colony ship headed to a planet called Eden. Someone is sabotaging the ship, and the craft's computer thinks you can stop it. Well, if you can, you win. Then the second part, "Return to Eden," starts with the crew of Snowball 9 waking and charging you with treason (a damaged tape in the control room seems to show that you attempted to blow up the ship). You escape to the planet's surface, and your second adventure begins. "Worm in Paradise," the final episode in this trilogy, takes place much later. You have become a sort of legend, a character of almost mythic proportions. (This information will be very valuable.)

Jewels of Darkness begins with "Colossal Adventure," the classic text adventure. You must find and explore the Colossal Caverns, then store the treasures you find there in a small cabin where you begin the game. After this challenge comes "Adventure Quest," where you must find and destroy the evil Demon Lord AGALIAREPT. The third game, "Dungeon Adventure," begins after you accomplish this difficult task, and you must explore the Demon Lord's domain for treasure, magic items, and adventure.

Although the plots that comprise *Silicon Dreams* are not familiar to me, the first one in *Jewels of Darkness*

certainly is. It is a variation of the original adventure. This program (the original, not Rainbird's version) is on many networks and is in the public domain (without pictures, of course). Some extra commands have been added, so Rainbird provides an improved version. There are undoubtedly many people who have not been playing computer games very long and who have never seen this classic before. It's still a fun game, and its historical importance (if it hadn't been for "Colossal Adventure," there probably wouldn't have been any others) makes it even more enjoyable.

The games are, in at least one sense, a good value. The plots and text-descriptions of the settings are done well enough, and you get three separate games for the price of one. These (along with the historical and nostalgic aspects) are about all the positive things you can say about the products. The graphics are the weakest point, which is surprising, since Rainbird had such good graphics in their other graphic text adventure, *The Pawn*. I notice that *The Pawn* was created by a different company than *Jewel of Darkness* and *Silicon Dreams*, so perhaps that explains the difference.

Although the games are challenging and the stories are interesting, the graphics are so far below par that they really detract from the gaming experience. Today's computer user (especially the Amiga owner) is accustomed to graphics that have at least eight colors (we expect at least

(continued)

COMPUTER MART

**Your Texas Amiga Source
Immediate Access to over 400 Amiga Titles.
Prices too low to print!**

**We stock Amiga Software and Peripherals
For A500, A1000, & A2000.
Mon. thru Fri. 10:00 AM-7:00 PM, Sat. 12:00-5:00 PM**

**CALL TOLL FREE
800-443-8236**

**CUSTOMER SERVICE
409-560-2826**

Computer Mart • 105 Lynn Street • Nacogdoches, TX 75961

twice that, but those still in the eight-bit world are not so fortunate). We are so used to good graphics that these games might actually startle some people. (There was a time when this was the best that computerdom had to offer!) The system lets you turn off the pictures via a menu selection. I tried playing the games this way, but then I was playing the "Colossal Adventure" that is on Compuserve! The parser is primitive (it was written a while ago), the descriptions are colorful at points, and the syntax has been expanded (Ramsave and OOPS! come to mind as the most useful new features). But this is not enough to live up to the our expectations. The games need the support of graphics, and the graphics can't even support themselves.

Rainbird is to be commended for one thing, however. The back of the package (which appears to be a videocassette box) has real pictures

from the game. They are not touched up or artificially enhanced. They look as bad on the package as they do on the screen. At least they are not hiding anything or giving you a false impression. The way marketing is today, this honesty is a big deal, and Rainbird deserves credit for it.

The new features of Ramsave and OOPS! deserve credit also. It is nice to be able to save your present position in RAM quickly. It also frees you from the copy protection scheme these games share. It is documentation-based, and I'm all for it. It comes up when you try to restore from a disk-based save. This can get a little tedious if you are doing a lot of restoring. Ramsave is very good for this type of situation.

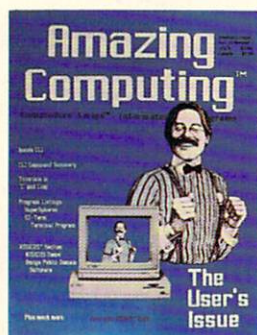
OOPS! is equally utilitarian. It allows you to take back your last move, no matter how disastrous it was. This makes exploring dangerous places very easy. It is an excellent feature,

and I would be pleased if other companies in the text adventure business tried to implement it.

There are some very good things you can say about these games. Players who are not that interested in graphics, or who find the history of gaming interesting, will especially enjoy these inexpensive games. The rest might wish to pass them by. Certainly, observant players won't be surprised by what they get. For the money, three old games punched up to mediocrity is not a bad deal. If nothing else, they are a good training ground for those of you just getting started in a career of adventuring. Many of us got our first taste of dungeoning with these very games. Just don't expect graphics like those in *The Pawn*, or even *Leisure Suit Larry in the Land of the Lounge Lizards*. Expect to be taken back to the days when the home computer industry was young, and four colors were a lot.

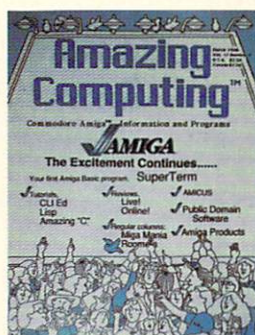
•AC•

Expanding Reference



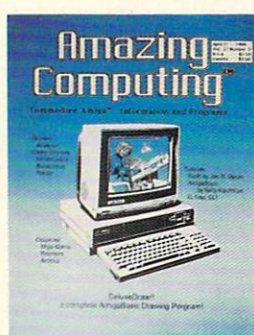
Volume 1 Number 1 Premiere 1986

Super Spheres By Kelly Kaufman An ABASIC Graphics prog.
Data Virus By J. Foust A disease may attack your Amiga!
EZ-Term By Kelly Kaufman An ABASIC Terminal program
Mega Mania By P. Kivolowitz Programming fixes & mouse care
Inside CLI By G. Musser a guided insight into the AmigaDOS™
CLI Summary by G. Musser Jr. A list of CLI commands
AmigaForum by B. Lubkin Visit Compuserve's Amiga SIG
Commodore Amiga Development Program by D. Hicks
Amiga Products A listing of present and expected products



Volume 1 Number 2 March 1986

Electronic Arts Comes Through A review of software from EA
Inside CLI: part two G. Musser Investigates CLI & ED
A Summary of ED Commands
Live! by Rich Miner A review of the Beta version of Live!
Online and the CTS Fabrite 2424 ADH Modem by J. Foust
Superterm V 1.0 By K. Kaufman A term. prog. in Amiga Basic
Amiga Bench "More" Program by Rick Wirth
Amiga BBS numbers



Volume 1 Number 3 April 1986

Analyze! a review by Ernest Viveiros
Reviews of Racter, Barstacass and Mindshadow
Forth! The first of our on-going tutorial
Deluxe Draw! by R. Wirth An Amiga Basic art program
Amiga Basic, A beginners tutorial
Inside CLI: part 3 by George Musser George gives us PIPE



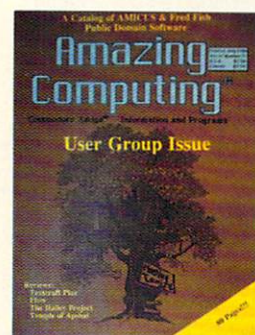
Volume 1 Number 4 May 1986

SkyFox and Artisoft Reviewed
Build your own 5 1/4 Drive Connector By Ernest Viveiros
Amiga Basic Tips by Rich Wirth
Scrimper Part One by P. Kivolowitz prog to print Amiga screen
Microsoft CD ROM Conference by Jim O'Keane
Amiga BBS Numbers



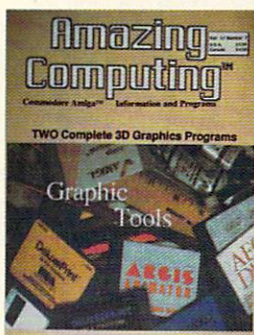
Volume 1 Number 5 1986

The HSI to RGB Conversion Tool
by S. Pietrowicz Color manipulation in BASIC
AmigaNotes by Rick Rase The first of the Amiga music columns
Sidecar A First Look by John Foust A first "under the hood"
John Foust Talks with R. J. Mical at COMDEX™
How does Sidecar affect the Transformer
an interview with Douglas Wyman of Smile
The Commodore Layoffs by J. Foust A look Commodore "cuts"
Scrimper Part Two by Perry Kivolowitz
Marauder reviewed by Rick Wirth
Building Tools by Daniel Kary



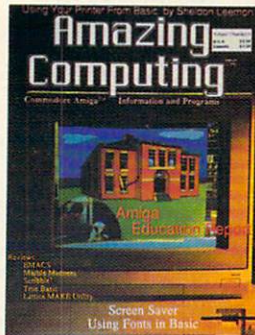
Volume 1 Number 6 1986

Temple of Aphel Trilogy reviewed by Stephen Pietrowicz
The Halley Project: A Mission in our Solar System
reviewed by Stephen Pietrowicz
Flow: reviewed by Ery Bobo
Textcraft Plus a First Look by Joe Lowery
How to start your own Amiga User Group by William Simpson
Amiga User Groups
Mailing List by Kelly Kaufman a basic mail list program
Pointer Image Editor by Stephen Pietrowicz
Scrimper: part three by Perry Kivolowitz
Fun With the Amiga Disk Controller by Thom Sterling
Optimize Your AmigaBasic Programs for Speed by Pietrowicz



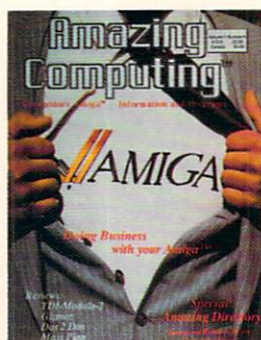
Volume 1 Number 7 1986

Angia Draw: CAD comes to the Amiga by Kelly Adams
Try 3D by Jim Meadows an introduction to 3D graphics
Angia Images/ Animator: a review by Ery Bobo
Deluxe Video Construction Set reviewed by Joe Lowery
Window requesters in Amiga Basic by Steve Michel
ROT by Colin French a 3D graphics editor
"I C What I Think" Ron Peterson with a few C graphic progs
Your Menu Sir! by B. Casey program Amiga Basic menus
IFF Brush to AmigaBasic "BOB" Basic editor by M. Swinger
Linking C Programs with Assembler Routines on the Amiga
by Gerald Hull



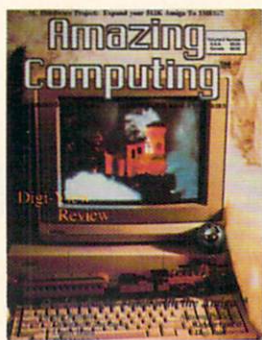
Volume 1 Number 8 1986

The University Amiga By G. Gamble
Amiga's inroads at Washington State University
MicroEd a look at a one man army for the Amiga
MicroEd, The Lewis and Clark Expedition reviewed Frizelle
Scribble Version 2.0 a review
Computers in the Classroom by Robert Frizelle
True Basic reviewed by Brad Grier
Using your printer with the Amiga
Marble Madness reviewed by Stephen Pietrowicz
Using Fonts from AmigaBasic by Tim Jones
Screen Saver by P. Kivolowitz A monitor protection prog. in C
Lattice MAKE Utility reviewed by Scott P. Evernden
A Tale of Three EMACS reviewed by Steve Poling
Jmpap File Reader in Amiga Basic by J. Jones



Volume 1 Number 9 1986

Instant Music Reviewed by Steve Pietrowicz
Mindwalker Reviewed by Richard Krepper
The Alegra Memory Board Reviewed by Rich Wirth
TxEd Reviewed by Jan and Cliff Kent
Amazing Directory A guide to the sources and resources
Amiga Developers A listing of Suppliers and Developers
Public Domain Catalog A listing of Amicus and Fred Fish PDS
Doe 2 Doe review R. Krepper
Transfer files from PCMS-DOS and AmigaBasic
MaxPlan review by Richard Krepper The Amiga Spreadsheet
Gizmox reviewed by Peter Wayner Amiga extra!
The Loan Information Program by Brian Catley
basic prog. to for your financial options
Starting Your Own Amiga Related Business by W. Simpson
Keep Track of Your Business Usage for Taxes by J. Kummer
The Absort Amiga Fortran Compiler reviewed by R. A. Reale
Using Fonts from AmigaBasic, Part Two by Tim Jones
68000 Macro on the Amiga by G. Hull Advance your ability.
TDI Modula-2 Amiga Compiler review by S. Fawcett



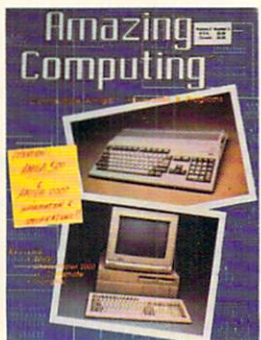
Volume 2 Number 1 1987

What Digi-View Is... Or, What Genlock Should Be by J. Foust
AmigaBasic Default Colors by Bryan Catley
A Public Domain Module-2 System reviewed by Warren Block
One Drive Compile by Douglas Lovell
Using Lattice C with a single drive system
A Megabyte Without Megabyte by Chris Irving
An Internal Megabyte upgrade
Digi-View reviewed by Ed Jakob
Defender of the Crown reviewed by Keith Conforti
Leader Board reviewed by Chuck Raudonis
Roundhill Computer System's PANEL reviewed by Ray Lane
Digi-Paint... by New Tek reviewed by John Foust
Deluxe Paint II... from Electronic Arts reviewed by J. Foust



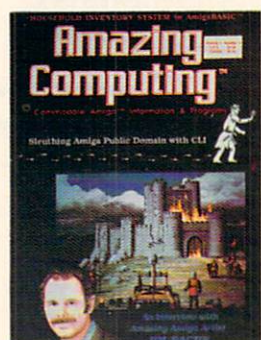
Volume 2 Number 2 1987

The Modem by Joseph L. Rothman efforts of a BBS Sysop
MacroModem reviewed by Stephen R. Pietrowicz
GEMINI or "It takes two to Tango" by Jim Meadows
Gaming between machines
BBS-PC reviewed by Stephen R. Pietrowicz
The Trouble with Xmodem by Joseph L. Rothman
The ACO Project... Graphic Teleconferencing on the Amiga
by S. R. Pietrowicz
Flight Simulator III: A Cross Country Tutorial by John Rafferty
A Disk Librarian in AmigaBASIC by John Kennan
Creating and Using Amiga Workbench Icons by C. Hansel
AmigaDOS version 1.2 by Clifford Kent
The Amazing MIDI Interface build your own by Richard Rae
AmigaDOS Operating System Calls and
Disk File Management by D. Haynie
Working with the Workbench by Louis A. Mamakos Prog in C



Volume 2 Number 3

The Amiga 2000™ by J. Foust
A First look at the new, high end Amiga™
The Amiga 500™ by John Foust
A look at the new, low priced Amiga
An Analysis of the New Amiga PCs by J. Foust
Speculation on the New Amigas
Gemini Part II by Jim Meadows
The concluding article on two-player games
Subscripts and Superscripts in AmigaBASIC by Ivan C. Smith
The Winter Consumer Electronics Show by John Foust
AmigaTrix by W. Block Amiga™ shorts
Intuition Gadgets by Harriet Maybeck Tolly
A journey through gadget-land, using C
Shanghai reviewed by Keith M. Conforti
Chessmaster 2000 & Chessmate reviewed by Edwin V. Apel, Jr.
Zing! by Meridian Software reviewed by Ed Bercovitz
Forth! by Jon Bryan Get stereo sound into your Forth programs.
Assembly Language on the Amiga™ by Chris Martin
Roomers by the Bandito Genlocks are finally shipping, & MORE!!!
AmigaNotes by R. Rae Hum Busters... "No stereo? Y not?"
The AMICUS Network by J. Foust
CES, user group issues and Amiga Expo*



Volume 2 Number 4 1987

Amazing Interviews Jim Sachs by S. Hull Amiga Artist
The Mouse That Got Restored by Jerry Hull and Bob Rhode
Sleuthing Public Domain Disks with CLI by John Foust
Highlights from the San Francisco Commodore Show
by S. Hull
Speaker Sessions: San Francisco Commodore Show H Tolly
The Household Inventory System in AmigaBASIC™
by B. Catley
Secrets of Screen Dumps by Nathan Olson
Using Function Keys with MicroEmacs by Greg Douglas
AmigaTrix II by Warren Block More Amiga shortcuts
Basic Gadgets by Brian Catley Create gadget functions
Gridiron reviewed by K. Conforti Real football for the Amiga
Star Fleet I Version 2.1 reviewed by J. Tracy Amiga Space
The TIC reviewed by J. Foust Battery powered Clock Calendar
Metascope review by H. Tolly An easy-to-use debugger



Volume 2 Number 5 1987

The Perfect Sound Digitizer review by R. Battle
The Future Sound Digitizer by W. Block Applied Vision's SD
Forth! by J. Bryan comparing JForth and MUI-Forth
Basic Input by B. Catley AmigaBASIC input routine for use in
all your programs
Writing a SoundScape Module in C by T. Fay Programming
with MIDI, Amiga and SoundScape by SoundScape author
Programming in 68000 Assembly Language by C. Martin
Continuing with Counters & Addressing Modes
Using FutureSound with AmigaBASIC by J. Meadows
AmigaBASIC Programming utility with real, digitized STEREO
AmigaNotes by R. Rae A review of Mimetec's
SoundScape Sound Sampler
More AmigaNotes by R. Rae
A further review of Sunrise's Perfect Sound
Waveform Workshop in AmigaBASIC by J. Shields edit & save
waveform for use in other AmigaBASIC programs
The Mimetec Pro MIDI Studio by Sullivan, Jeffery
A review of Mimetec's music editor/player
Intuition Gadgets Part II by H. MaybeckTolly Boolean gadgets
provide the user with an on/off user interface.



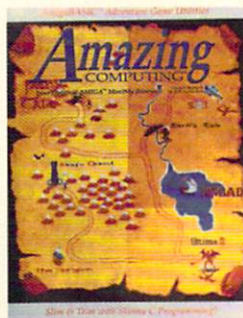
Volume 2 Number 6 1987

Forth! by J. Bryan Access: resources in the ROM kernel.
The Amazing Computing Hard Disk Review by J. Foust & S.
Leon In-depth looks at the C Ltd. Hard Drive, Microbot's
MAS-Drive20, Byte by Byte's PAL Jr., Supra's 4x4 Hard Drive
and Xebec's 9720H Hard Drive. Also, a look at disk driver
software currently under development.
Module-2 AmigaDOS™ Utilities by S. Fawcett A
Call to AmigaDOS and the ROM kernel.
Amiga Expansion Peripheral by J. Foust
Explanation of Amiga expansion peripherals.
Amiga Technical Support by J. Foust
How and where to get Amiga tech support.
Goodbye Los Gatos by J. Foust Closing Los Gatos.
The Amicus Network by J. Foust West Coast Computer Faire.
Metacomms Shell and Toolkit by J. Foust A review
The Magic Sac by J. Foust Run Mac programs on your Amiga.
What You Should Know Before Choosing an Amiga 1000
Expansion Device by S. Grant
7 Assemblers for the Amiga by G. Hull Choose your assembler
High Level Shakeup Replaces Top Management at
Commodore by S. Hull
Peter J. Bazzor by S. Hull Manager at CBM gives an inside look
Logix: A review by Richard Krepper
Organizer! by A review Richard Krepper database.
68000 Assembly Language Programming on the Amiga
by Chris Martin
Superbase Personal Relational Database by Ray McCabe
AmigaNotes by Rae, Richard A look at FutureSound
Commodore Shows the Amiga 2000 and 500 at the Boston
Computer Society by H. Maybeck Tolly



Volume 2, Number 7 1987

New Breed of Video Products by John Foust...
Very Vivid! by Tim Grantham...
Video and Your Amiga by Oran Sands III
Amigas & Weather Forecasting by Brendon Larson
A-Squared and the Live! Video Digitizer by John Foust
Aegis Animator Scripts and Cel Animation by John Foust
Quality Video from a Quality Computer by Oran Sands III
Is IFF Really a Standard? by John Foust
Amazing Stories and the Amiga™ by John Foust
All about Printer Drivers by Richard Belek
Intuition Gadgets by Harriet Maybeck Tolly
Deluxe Video 1.2 by Bob Eller
Pro Video CG1 by Oran Sands III
Digit-View 2.0 Digitizer/Software by Jennifer M. Janik
Prim HAM Editor from Impulse by Jennifer M. Janik
Easy! drawing tablet by John Foust
CSA's Turbo-Amiga Tower by Alfed Abuto
68000 Assembly Language by Chris Martin



Volume 2, Number 8 1987

This month Amazing Computing™ focuses on entertainment packages for the Amiga. Amazing game reviews...

SDI, Earl Weaver Baseball, Portal, The Surgeon, Little Computer People, Siroad, StarGlider, King's Quest (I, II and III), Fairy Tale Adventure, Ultima III, Facets of Adventure, Video Vegas and Bard's Tale.

Plus Amazing monthly columns... Amiga Notes, Rooms, Modula-2, 68000 Assembly Language and The Amicus Network.

Disk-2-Disk by Matthew Leeds
The ColorFont Standard by John Foust
Skinny C Programs by Robert Riemersma, Jr.
Hidden Messages in Your Amiga™ by John Foust
The Consumer Electronics Show and Comdex by J. Foust



Volume 2, Number 12 1987

The Ultimate Video Accessory by Larry White
The Sony Connection by Stewart Cobb
15-Puzzle in AmigaBASIC by Zoltan Szepel
Life, Part I: The Beginning by Gerald Hall
The ultra-complex nine bit solution to the "Game of Life."

Amiga Virus! by John Foust

A new Amiga virus has surfaced. Please check your system.

CLI Arguments in C by Paul Castonguay

MIDI Interface Adapter by Barry Masson

Amiga 1000-style MIDI interfaces can fit A2000s or 500s

Modula-2 by Steve Fawcett

First in a series, a command line calculator in Modula-2.

AmigaNotes by Richard Rae

The audio changes made in the Amiga 500 and 2000.

Animation for C Rookies: Part III by M. Swinger

tackling double-buffering.

The Big Picture by Warren Ring

Amiga™ Assembly language programming for the brave!

Karate Kid Review by Stephen R. Pietrowicz

GO! 64 review by John Foust, James O'Keane, and Rick Wich

Three C-64 experts investigate a new Amiga 64 emulator.

A-Talk-Plus Review by Brendan Larson

"Full-fledged terminal program" & Tektronics capabilities

Calligrapher Review by John Foust

Animator: Apprentice Review by John Foust

Playing Dynamic Drums on the Amiga by David N. Blank

WordPerfect Review by Steve Hull

Insider/Kwikstart Review by Ernest P. Viveiros

RAM & ROM expansion: Comments and installation tips.

Bug Bytes by John Steiner

Forth! by Jon Bryan

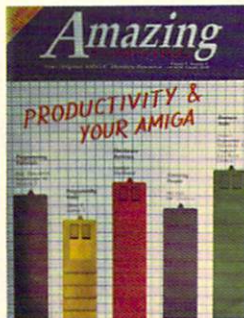
DumpPort utility for your Multi-Forth toolbox.

As I See It by Eddie Churchill

An offbeat look on Digi-Paint, Portal, and Videopace 3D.

The Amicus Network by John Foust

The Commodore Show and AmExpo: New York



Volume 2, Number 9 1987

Analyze 2.0 reviewed by Kim Schaffer
Impact Business Graphics review by Chuck Raudonis

Microfiche Filter review by Harv Laser

Pagesetter review by Rick Wirth

Gizmo Productivity Set 2.0 review by Bob Eller

Kickwork review by Harv Laser

Digs Telecommunications Package review by Steve Hull

Mouse Time and Timesaver review by John Foust

Insider Memory Expansion review by James O'Keane

Microbotics Starboard-2 review by S. Fawcett

Leather Goddesses of Phobos

reviewed by Harriet Maybeck-Tolly

Lattice C Compiler Version 3.10 reviewed by Gary Sarff

Manx 3.4a Update reviewed by John Foust

AC-BASIC reviewed by Sheldon Leemon

AC-BASIC Compiler an alternative comparison by B. Catley

Modula-2 Programming S Fawcett

Raw Console Device Events

Directory Listings Under AmigaDOS by Dave Haynie

AmigaBASIC Patterns by Brian Catley

Programming with Soundscapes

Todor Fay manipulates the samples

Bill Volk, Vice-President Aegis Development,

interviewed by Steve Hull

Jim Goodnow, Developer of Manx 'C'

interview by Harriet M. Tolly



Volume 2, Number 10 1987

Max Headroom and the Amiga by John Foust
Taking the Perfect Screen Shot by Keith Conforti

Amiga Artist: Brian Williams by John Foust

Amiga Forum on Compuserve™... Software Publishing

Conference Transcript by Richard Rae

All About Online Conferencing by Richard Rae

dBMAN reviewed by Clifford Kent

Amiga Pascal reviewed by Michael McNeil

AC-BASIC Compiler reviewed by Bryan Catley

Bug Bytes by John Steiner

Amiga Notes by Richard Rae

Roomers by The Bandito

68000 Assembly Language by Chris Martin

The AMICUS Network by John Foust

Amiga Programming:

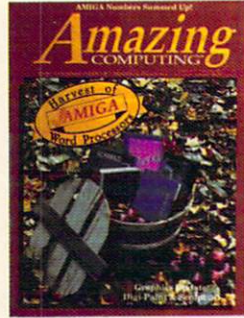
Amiga BASIC Structures by Steve Michel

Quick and Dirty Bobs by Michael Swinger

Directory Listings Under Amiga-DOS, Part II by Dave Haynie

Fast File I/O with Modula-2 by Steve Fawcett

Window I/O by Read Predmore



Volume 2, Number 11 1987

Word Processors Rundown by Geoff Gamble

ProWrite, Scribble, and WordPerfect compared

LPD Writer Review by Marion Deland

Vista Writer Review by Harv Laser

Audit Review by Warren Book

WordPerfect Review by Harv Laser

Jez San Interview by Ed Berovitz

The author of StarGlider speaks!

Do-it-yourself Improvements to the Amiga Genlock

Digi-Paint Review by Harv Laser

Sculpt 3D Review by Steve Pietrowicz

Shadowgate Review by Linda Kaplan

TeleGames Review by Michael T. Cabral

Reason Preview: a quick look at an

intense grammar examination application

As I See It by Eddie Churchill: Peeking at WordPerfect,

Gizmo V2.0 and Zing! Keys

Bug Bytes by John Steiner

AmigaNotes by R. Rae 4 electronic music books

Modula-2 Programming by Steve Fawcett

devices, I/O, and the serial port

Roomers by The Bandito

68000 Assembly Language by Chris Martin

Chris walks through the display routines

The AMICUS Network by John Foust

Desktop Publishing & Seybold

C Animation Part II by Mike Swinger Animation Objects

BASIC Text by Brian Catley Pixel perfect text positioning

Soundscapes Part III by Todor Fay VU Meter and more

Fun with Amiga Numbers by Alan Barnett

File Browser by Bryan Catley

Full Feature BASIC File Browsing utility

Your Resource to the Commodore Amiga™

The phrase above is not just empty words. The pages of Amazing Computing™ are filled with articles on technical operations and procedures, basic use, and just-plain-fun. The growing library of Amazing Back Issues contains articles from building your own IBM Disk controller, to setting up your own startup sequence. Amazing Computing™ has repeatedly been the first magazine to offer the Amiga users solid, indepth reviews and hands on articles for their machines.

Amazing Computing™ was the first magazine to document CLI
Amazing Computing™ was the first to show Sidecar™ from COMDEX™ in full detail.
Amazing Computing™ was the first to document a 5 1/4 drive connector
Amazing Computing™ was the first with a 1 Meg Amiga upgrade hardware project!
Amazing Computing™ was the first magazine to offer serious programming examples and help.
Amazing Computing™ was the first to offer Public Domain Software at reasonable prices.
Amazing Computing™ was the first magazine with the user in mind!

From the Beginning

Since February 1986, Amazing Computing™ has been providing users with complete information for their Amiga. This store house of programs and information is still available through our back issues. From the Premiere issue to the present, there are insights into the Amiga that any user will find useful.

\$4.00 each!

Our Back Issue price is still \$4.00 per issue! (Foreign orders, please add \$1.00 per issue for Postage & Handling. All payments must be made by check or money order in U.S. funds drawn on a U.S. Bank.)

Limited Supply

Unfortunately, nothing lasts forever, and the availability of some of our Back Issues is definitely limited. Please complete your Amazing Computing™ library today, while these issues are still available. Please complete the order form in the rear of this issue and mail with check or money order to:

Back Issues, PiM Publications, Inc., P.O. Box 869, Fall River, MA 02722
(Please allow 4 to 6 weeks for delivery)

Leisure Suit Larry in the **LAND** of the **LOUNGE** **LIZARDS**

"Don't expect scenes that would win an X-rating, but this one is definitely not for the kiddies."

reviewed by Kenneth E. Schaefer

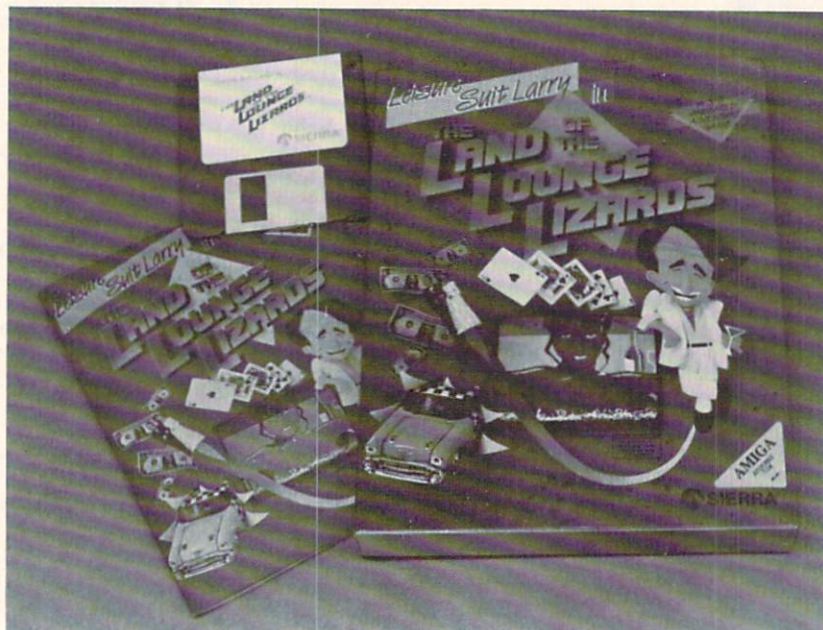
The singles' bar. It's a well-known place for many Americans. To many, spending an evening in such a bar is a good time. To others, a singles' bar is the bloody battlefield where the weary war against loneliness is futilely fought. To yet others, it is a place of evil and sin weakening the very fabric of our society. In fact, the only thing you can say with any degree of confidence about the singles' scene is that everyone has an opinion about it. So you can just imagine the noise that erupted when Sierra On-line introduced Leisure Suit Larry in the Land of the Lounge Lizards. When the game is played, the noise gets even louder. For many, however, that noise is not roars of indignation, but howls of laughter.

is unfortunate that such a good satire on contemporary American life is marred by such one-sidedness. Granted, it will probably take a completely separate game to remove the male bias from Leisure Suit Larry. What could the heroine of the female version be called? Evening Gown Evelian? Leisure Suit Louise? Be that

incensed. To those who are offended by what the game represents, I can only suggest that you skip the rest of this review.

Leisure Suit Larry in the Land of the Lounge Lizards is another game in the growing list of "Adult" software. For some games, this label is a bit

strained. For this one, it is not. Don't expect scenes that would win an X-rating, but this one is definitely not for the kiddies. Sierra has even attempted to install a "Child-protection scheme" to prevent people under 18 from playing. Whether this is a serious attempt at limiting access to the game, or another in a long line of jokes about age and responsibility, is anybody's guess. Only the authors know for sure. I personally doubt that it will keep



Make no mistake about it, Leisure Suit Larry in the Land of the Lounge Lizards is not for everyone. It most certainly isn't for young children. I suspect that it also isn't for women. It

as it may, most women probably will not find Leisure Suit Larry in the Land of the Lounge Lizards as funny or entertaining as many men will. In fact, many women will probably be

out determined young people. Parents will have to bear the final responsibility of keeping this software from eyes too young to "appreciate" it.

Our "hero" is Leisure Suit Larry, a quintessential nerd. Our goal is to help Larry become, shall we say, "more experienced" by directing his actions. In his quest for this rather shallow-fulfillment, Larry meets prostitutes and pimps, bouncers and bartenders, and even a loud-mouthed drug store clerk. He goes to casinos, discos, and seedy bars with even seedier backrooms. He can fall down fire escapes, be humiliated by dogs, and be mugged in dark alleys before attaining his goal—all this while battling what can only be called "chronic toxic breath syndrome." I guess some people just have it tough.

My only serious complaint about Leisure Suit Larry in the Land of the Lounge Lizards is that you cannot control the dialogue. (My biggest complaint is the name is *too long*!) You can only "Talk to Fawn" or "Talk to Faith." They then give you their programmed responses. This pattern of "Talk to ..." and response is repeated until you start getting the same answers over and over. You then know you've milked these people for all they know. This kind of interaction gives the game a "programmed" feel. It makes me feel there is only one way to get through the game (although I don't think that this is true). In the game's defense, "smart" people are very difficult to implement, and the game plays well without them.

Leisure Suit Larry in the Land of the Lounge Lizards uses the graphics and animation system that helped make the King's Quest series so popular. Some of these games have been ported to the Amiga, and I wasn't too impressed with the graphics in those earlier ports. Leisure Suit Larry in the Land of the Lounge Lizards appears to have taken care of this problem. The animation and graphics are up to an acceptable level. The only complaint you might have is that the size of the "world" might need to be shrunk to make room for better graphics. Leisure Suit Larry in the Land of the

Lounge Lizards does not seem as big as King's Quest III or Space Quest. The user-interface is very good, however, with keyboard equivalents for menu selections. In short, this game does not look like a straight IBM port.

Although the original premise of Leisure Suit Larry in the Land of the Lounge Lizards is hardly anything to write home about (Mom had better be *real* hip!), the story as it develops is funny and fun to play. Moral lessons can even be learned as you play. The most noticeable thing about Leisure Suit Larry in the Land of the Lounge Lizards, though, is the humor at every turn. Subtle things like getting an ad for Sierra's clue book when you use a pay phone to call Sierra Technical support make the game enjoyable to explore. Be sure to spend some time in the Cabaret when the comedian is on stage (although the Can-Can girls can be funny also). The puzzles, although difficult, are not impossible.

Leisure Suit Larry in the Land of the Lounge Lizards has a disk-based copy-protection scheme, so hard-drive users can forget transferring it into their Games directories. The disk, as it comes from Sierra, does not contain Workbench, so you have to boot off another disk. Leisure Suit Larry in the Land of the Lounge Lizards does, however, work correctly in expanded memory.

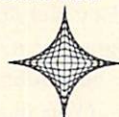
Overall, I found Leisure Suit Larry in the Land of the Lounge Lizards to be an enjoyable and challenging game. It is polished and free from those bothersome bugs that many companies just don't seem concerned enough to swat. Although the contents make it a limited-audience game (What game isn't?), I think those who are not offended by the game's premise will get hours of entertainment from it. Now, if you will excuse me, I have to go to the dry cleaner's to pick up my leisure suit.

•AC•

SOURCE BUILDER

Introducing — A CASE Tool for AMIGA
This powerful new Amiga CASE (Computer Aided Software Engineering) tool is for software designers to automate the creation and maintenance of software from cradle to grave.

- An effective, affordable, professional CASE tool for automated code generation.
- Easily build user interface and complex data structure routines. Supports reusable code and object oriented design.
- Logical template substitution for multiple version code generation; i.e. From one common form, generate code for multiple computers, PC, MAC, Amiga & ST.
- Extremely effective in non programming environment; CAD, Wordprocessing, Project Management, and Postscript processing for desk top publishing.
- Allows many ADA tm features to be used in conventional languages such as C. Also extends the object oriented design features of ADA tm.
- Templates and configurations can be easily added and expended by user.
- Includes an user manual as well as a software design tutorial.
- Price \$149.95, send check or money order only. CA residents add 6.5% sales tax. This product requires Amiga Workbench 1.2 and Extra disk.



Software Integration Solutions
16496 Bernardo Center Drive
San Diego, CA. 92128
Tel: 619-451-3094

More Basic Text

by Bryan Catley

When I wrote "Basic Text" for a recent issue of *Amazing Computing* (AC V 2.11), I didn't really expect to write a follow up, but it now seems appropriate. In that article, I discussed a number of different ways to display text on the screen, but I didn't package them. That's what I'll do now.

What I mean by packaging is making the routines convenient and easy to use. For example, if you want to use italics, you usually need to issue the command:

```
CALL SetSoftStyle&(WINDOW(8),4,255)
```

However, isn't the following "packaged" command much better?

```
SetStyle Italics
```

You don't need to read "Basic Text" to use the information in this article, but a minimal understanding of the following operating system routines will help: *Text&* (display a character string), *Move&* (position text cursor to a specific pixel), *SetSoftStyle&* (set italics, bold, etc.), and *SetDrMd&* (set JAM1, JAM2, etc.).

While I'm at it, I'll also provide some packaging for easy access to all disk fonts.

The Text Subprograms

Type in Listings #1 and #2 and save them in ASCII format, so they can be Merged into other programs. To do this, use the immediate command *SAVE "name,A."* The "A" is impor-

tant, so don't forget it! If you forget the "A" you'll get an error when you try to use the subprograms. If this happens, just *LOAD* and *re-SAVE* with the "A." (Name the listings "TextVars" and "Text," respectively.)

Listing #1 (TextVars) is a set of variables (used by both the *Text* and *Fonts* subprograms) which are pre-set to specific values, allowing you to say "italics" rather than "4" to produce italics. They should be Merged into your program at its beginning, or wherever you define your other variables. (You may want to leave the comments out when you enter this listing—they're only there for easy identification.)

Listing #2 (Text) is a collection of four subprograms allowing you to do just about anything with text! The subprograms are *SetStyle*, *SetMode*, *At*, and *Display*. Let's take a look at each.

SetStyle uses the *SetSoftStyle&* operating system routine and allows you to change your text to bold, italics, underline, or any combination thereof. Valid styles are "standard," "underline," "bold," and "italics." Combinations may be achieved by "adding" the desired styles. Examples:

```
SetStyle bold
SetStyle standard
SetStyle bold+italics
SetStyle standard+underline
```

SetMode uses the *SetDrMd&* operat-

ing system routine and allows you to change the display mode. Valid modes are "JAM1," "JAM2," "complement," and "inversvid." (See the original "Basic Text" article for an in-depth discussion of these modes.) Once again, these modes may be combined. Examples:

```
SetMode JAM1
SetMode JAM2+complement
Setmode inversvid
```

At uses the *Move&* operating system routine and allows you to position text at any pixel location on the screen. Don't forget, when you use *Move*, you must specify the x and y left edge coordinates of the text string's baseline. (With the standard display format, this is the seventh pixel from the top of the character.) This can become very confusing, especially when you are using disk-based fonts, so *At* takes care of it all for you; just specify the coordinates of the upper left-hand corner of the first character! Examples:

```
At 35,47
At 122,150
At 305,180
At x%,y%+5
```

Note: Because of the method *At* uses to compensate for the baseline, vertical positioning may be off by one pixel in either direction. In most cases, this discrepancy will not cause a problem, but if it does, just adjust the y coordinate accordingly.

Display uses the *Text&* operating

system routine and displays a given character string at the position set by a previous At command. Text& is preferable to Print when using any print style other than standard. It is also faster than Print. Examples:

```
Display "Hello, World!"
Display TextString$
Display "You "+x$+" person!"
```

Remember, Display is not a Print command! You cannot use semicolons. *Every Display must be preceded by an At command.* There is no automatic data conversion. In other words, you must format the string yourself before Displaying it (although you may use the "+" to concatenate several strings).

When used together, the four subprograms give you direct, easy control over your text displays. They also make your programs much easier to read. Consider the following combination for readability:

```
SetStyle bold+italics: At 30,50
Display "Good morning!"
```

There are two other things to remember. In order to use these subprograms, you must have a "graphics.bmap" file in the current directory (one is already in the BasicDemos drawer on your Amiga-BASIC disk), and you must issue a Library "graphics.library" statement before you use any subprograms.

The Font Subprograms

The default font, "topaz," is automatically available every time you use your Amiga. However, a number of other fonts stored on the Workbench disk are readily available. Let's take a look at three subprograms which make these fonts really easy to use. All you need to know is the name and height of the font you wish to use!

Type in Listings #3 and #4 and save them in ASCII format. Name them "FontVars" and "Fonts." Listing #3 (FontVars) contains variables and function declarations required by the subprograms in Listing #4 (Fonts). Listing #3 also contains a list of all available fonts, their heights, and the location of their baselines. For convenience, this list is reproduced here:

Font	Heights	Baseline at
diamond	12, 20	+9, +14
emerald	17, 20	+12, +14
garnet	9, 16	+7, +11
opal	9, 12	+7, +9
ruby	8, 12, 15	+6, +9, +11
sapphire	14, 19	+11, +15
topaz	8, 9, 11	+6, +6, +8

Notes:

Topaz8 and topaz9 are system-based and do not require loading from disk. These two fonts are also the default fonts, depending on whether the Amiga is in 80 or 60 column mode, respectively.

Remember, don't worry about the baseline because the At subprogram looks after it for you.

While you may use "TextVars" and "Text" alone, you will probably not use "FontVars" and "Fonts" without the other two!

The text subprograms require a "graphics.bmap" and a LIBRARY "graphics.library" statement, so the font subprograms require a "diskfont.bmap"v and a LIBRARY "diskfont.library" statement. Unfortunately, a "diskfont.bmap" does not come with AmigaBASIC. If you have not acquired one from the Public Domain, enter Listing #6, save it, and run it. This listing creates a "diskfont.bmap" file in the "Basic-Demos" drawer on your AmigaBASIC disk.

The three subprograms in "Fonts" are LoadFont, UseFont, and EndFont.

LoadFont loads the named font from disk (if necessary). A pointer to the font structure in memory is returned and is required by the other two subprograms. *Note:* LoadFont does not actually cause the named font to be used. Examples:

```
LoadFont "Ruby",12,RubyPtr&
LoadFont
FontName$,FontHeight$,FontPtr&
```

The first two parameters, font name and height, must be completed by the user prior to the call. "LoadFont" returns the font structure pointer in the third parameter.

UseFont specifies which font (the font's structure is pointed to by the single parameter) will be used in all subsequent Print and Display statements. Example:

```
UseFont FontPtr&
```

The value in FontPtr& is set by an earlier call to "LoadFont." *Note:* "UseFont" establishes the valid styles for the new font and automatically makes them known to "SetStyle."

EndFont specifies which font is unavailable for further use. Example:

```
EndFont FontPtr&
```

The value in FontPtr& is set by an earlier call to "LoadFont."

Using the Font Subprograms

There are two ways of using the three font subprograms. You can load all the fonts to be used during program initialization, issue "UseFont" statements as required, and close all fonts at program termination. You can also issue "LoadFont," "UseFont," and "EndFont" statement combinations as each font is required. Generally speaking, the first method is the best because it does not delay the

(continued)

5 Reasons Why You're Ready For MacroModem

1. You love telecom, but not memorization. MacroModem's user-written macro libraries and companion help screens (36 macros per file) store log on procedures, remote system menus and commands,
2. You've always wanted to use the mouse after you're connected, too. Write macros that mimic remote system commands and menus, then execute them with the mouse or keyboard.
3. You like automation, but not script languages. Our macros use normal commands from MacroModem, remote systems, and AmigaDOS, as well as text and control codes. A multi-windowed MacroEditor is included. No new programming language to learn.
4. You want to do other things while downloading a file. MacroModem is truly multi-tasking, with a NewCLI available anytime, even during file transfers. And MacroModem's error checking won't stop downloads unless you tell it to.
5. Of course MacroModem includes standard telecom software features, too. Teach MacroModem what you want, and it will remember for you.

MacroModem - the better way to do telecommunications. \$69.95

Kent Engineering & Design
P.O. Box 178, Mottville, NY 13119
(315) 685-8237

program's execution each time a new font is needed. However, it does require more memory. The choice is yours!

Putting It All Together

Examples of all subprograms (Text and Font) are included in Listing #5 which, when entered and run, show some things you can do with plain old Amiga text! You may be surprised. Type it in, merging the files you saved earlier as instructed in the listing.

To merge one program with another, type the immediate command Merge "name." Program "name" is read from disk and appended to the current program. If you get an error message, you probably forgot to save the files in ASCII format. Reload each and save it again; this time with the ",A" suffix. Your program's text output need not look dull and ordinary any more!

```
' Listing #1 - TextVars
' 'More Basic Text' by Bryan D. Catley
'
' *****
' *** These variables and this library ***
' *** statement are used in ***
' *** conjunction with the 'Text' ***
```

```
' *** and 'Fonts' collection of ***
' *** sub-programs. ***
' *****
standard=0:underline=1:bold=2:italics=4
JAM1=0:JAM2=1:complement=2:inversvid=4
ValidStyles%=255:FontHeight%=8
CHDIR":BasicDemos"
LIBRARY "graphics.library"
CHDIR ":"

' *****

' Listing #2 - Text
' 'More Basic Text' by Bryan D. Catley
'
' *****
' *** These sub-programs are used ***
' *** in conjunction with the set ***
' *** of 'TextVar' variables. ***
' *****

SUB SetStyle (style) STATIC
  SHARED ValidStyles%
  style%=style
  CALL SetSoftStyle%(WINDOW(8),style%,ValidStyles%)
END SUB

SUB SetMode (mode) STATIC
  mode%=mode
  CALL SetDrMd%(WINDOW(8),mode%)
END SUB

SUB At (x%,y%) STATIC
  SHARED FontHeight%
  x%=x%:y%=y%+INT(FontHeight%*.75)
  CALL Move%(WINDOW(8),x%,y%)
END SUB

SUB Display (Txt$) STATIC
  CALL Text%(WINDOW(8),SADD(Txt$),LEN(Txt$))
END SUB
```

```
' Listing #3 - FontVars
' 'More Basic Text' by Bryan D. Catley
'
' *****
' *** These variables, library, ***
' *** and declare statements are ***
' *** used in conjunction with ***
' *** the 'Fonts' collection of ***
' *** subprograms. ***
' *****

' Fonts distributed with the Amiga are:
'
' Font      Heights      Baseline at
' -----
' diamond   12, 20      +9, +14
' emerald   17, 20      +12, +14
' garnet     9, 16       +7, +11
' opal       9, 12       +7, +9
' ruby       8, 12, 15   +6, +9, +11
' sapphire  14, 19       +11, +15
' topaz      8, 9, 11    +6, +6, +8
'
' Note: Topaz 8 and 8 are system-based
' and do not require loading from disk.
'
DECLARE FUNCTION OpenFont% LIBRARY
DECLARE FUNCTION OpenDiskFont% LIBRARY
DECLARE FUNCTION AskSoftStyle% LIBRARY
DIM textAttr%(1)
CHDIR":BasicDemos"
LIBRARY "diskfont.library"
CHDIR ":"
```



```

' Listing #4 - Fonts
' 'More Basic Text' by Bryan D. Catley
'
' *****
' *** These sub-programs are used ***
' *** in conjunction with the set ***
' *** of "FontVars" and ***
' *** "TextVars" variables. ***
' *****
'
SUB LoadFont (FontName$,FontHeight$,FontPtr%) STATIC
  SHARED FontHeight$,textAttr%
  FontHeight%=FontHeight$
  textAttr% (0)=SADD (FontName$+".font"+CHR$(0))
  textAttr% (1)=FontHeight$*65536%
  IF (FontName$="topaz") AND (FontHeight%=8 OR FontHeight%=9)
  THEN
    FontPtr%=OpenFont% (VARPTR (textAttr% (0)))
  ELSE
    FontPtr%=OpenDiskFont% (VARPTR (textAttr% (0)))
  END IF
END SUB

```

```

SUB UseFont (FontPtr%) STATIC
  SHARED ValidStyles%
  CALL SetFont% (WINDOW (8),FontPtr%)
  ValidStyles%=AskSoftStyle% (WINDOW (8))
END SUB

```

```

SUB EndFont (FontPtr%) STATIC
  CALL CloseFont% (FontPtr%)
END SUB

```

```

Listing #5 - Text and Font Demonstration
' 'More Basic Text' by Bryan D. Catley
'
' A demonstration of the "Text" and "Fonts"
' sub-programs described in the accompanying
' article.
'
' Now issue the two immediate commands:
'
' MERGE "TextVars"
' MERGE "FontVars"
'
' Now continue with the main program...
'

```

```

SCREEN 2,640,200,3,2:WINDOW 2,,,16,2
PALETTE 0,.4,.1,0:Brw=0
PALETTE 1,0,0,1:Blu=1
PALETTE 2,0,0,0:Blk=2
PALETTE 3,1,0,1:Mag=3
PALETTE 4,1,1,0:Yel=4
PALETTE 5,0,1,0:Grn=5
PALETTE 6,1,0,0:Red=6
PALETTE 7,.5,.5,.5:Gra=7
COLOR ,Gra:CLS
LINE (144,0)-(496,1),Red,bf
LINE (136,2)-(504,3),Blu,bf
LINE (128,4)-(512,5),Yel,bf
LINE (120,6)-(520,7),Grn,bf
LINE (120,8)-(520,9),Grn,bf
LINE (128,10)-(512,11),Yel,bf
LINE (136,12)-(504,13),Blu,bf
LINE (144,14)-(496,15),Red,bf

```

```

Txt$="HOW'S THIS FOR A HEADING?"
SetMode JAM1:SetStyle italics
COLOR Blk:At 216,4:Display Txt$
COLOR Mag:At 218,4:Display Txt$

```

```

Txt$="C l i c k t o Q u i t !"
SetMode JAM1:SetStyle bold+italics
COLOR Yel:At 232,184:Display Txt$
COLOR Red:At 234,185:Display Txt$

```

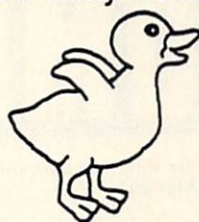
```

Txt$="STROBE style"

```

NEW FROM ROBOT READERS

A Powerful New Way To Learn To Read



The Ugly Duckling

ROBOT READERS Are Designed to
Teach Children To Read In An Effective,
Positive Way Never Before Possible

ALSO AVAILABLE

- | | |
|------------------|---------------------|
| * CHICKEN LITTLE | * AESOP'S FABLES |
| * LITTLE RED HEN | * THREE LITTLE PIGS |

SUGGESTED **\$29.95 each**
RETAIL: for the Amiga 512k

AT YOUR FAVORITE COMPUTER STORE

Hilton Android
PO Box 7437, Huntington Beach, CA 92615
(714) 963-4584

```

SetMode JAM1:SetStyle standard
row%=28:n=0
FOR column%=56 TO 74 STEP 3
  COLOR n:At column%,row%:Display Txt$
  n=n+1:row%=row%+1
NEXT

```

```

Txt$="Shadowed Text"
SetMode JAM1:SetStyle bold
COLOR Blk:At 216,28:Display Txt$+" 1"
COLOR Grn:At 218,29:Display Txt$+" 1"
COLOR Blu:At 232,44:Display Txt$+" 2"
COLOR Blk:At 235,45:Display Txt$+" 2"

```

```

SetMode JAM2+inversvid:SetStyle italics
COLOR Mag,Brw:At 410,32
Display " Note inversed Colors "

```

```

LoadFont "diamond",20,diamond20ptr%
UseFont diamond20ptr%
SetMode JAM1:SetStyle bold
COLOR Yel:At 40,64:Display "B i g and B o l d"
LoadFont "topaz",8,topaz8ptr%
UseFont topaz8ptr%
SetStyle standard
COLOR Red:At 72,68:Display "Striking effect!"

```

```

LoadFont "sapphire",14,sapl4ptr%
UseFont sapl4ptr%
SetStyle italics
COLOR Blk:At 240,80:Display "Sapphire 14 Italics"
SetStyle standard
COLOR Red:At 396,56:Display "Sapphire 14 Standard"

```

(continued)

GOMF!

"...it is small, installs with just a double-click on an icon...a welcome utility for Amiga users."

-INFO magazine

"...Without a doubt, the most important program this month is GOMF. All Amiga users are sick and tired of losing running programs and data to the dreaded Guru, and anything that can stop this terrible syndrome deserves a round of applause!"

"...So if Gurus are getting to you, add GOMF to your Amiga environment and you can start saying goodbye to the GURU too!"

-Commodore magazine

Now GOMF is even better!
Version 2.0 NOW shipping

NEW FEATURES include:

- Automatic removal of ALL errors (task held, GURU, etc.)
- Preserves low memory auto vectors.
- NUKE those pesky CANCEL requestors once and for ALL!
- Eliminate ANY task or device AT ANY TIME.
- auto-configures for 68000/68010/68020.
- CLI and Workbench compatible.
- Extensive, easy to understand documentation.

only \$39.95

```
UseFont diamond20ptr&
SetStyle bold
COLOR Blk:At 245,105:Display "More Shadows"
SetStyle standard
COLOR Grn:At 248,105:Display "More Shadows"
COLOR Blk:At 390,125:Display "...and still more!"
COLOR Yel:At 387,124:Display "...and still more!"
```

```
LoadFont "ruby",12,ruby12ptr&
UseFont ruby12ptr&
COLOR Brw:At 96,130:Display "Ruby 12 point"
```

```
LoadFont "emerald",17,emerald17ptr&
UseFont emerald17ptr&
COLOR Mag:At 75,155:Display "Emerald 17 point"
```

```
LoadFont "opal",12,opal12ptr&
UseFont opal12ptr&
COLOR Grn:At 360,150:Display "Opal 12 point"
```

```
LoadFont "garnet",9,garnet9ptr&
UseFont garnet9ptr&
COLOR Red:At 400,165:Display "Garnet 9 point"
COLOR Blk:At 402,166:Display "Garnet 9 point"
```

```
WHILE MOUSE(0)=0:WEND
EndFont diamond20ptr&
EndFont sap14ptr&
EndFont ruby12ptr&
EndFont emerald17ptr&
EndFont opal12ptr&
EndFont garnet9ptr&
SetMode JAM2
LIBRARY CLOSE
WINDOW CLOSE 2:SCREEN CLOSE 2
END
```

New Products for All **AMIGAS**
from **HyperTek/Silicon Springs**

Deluxe MIDI interface

Simply the BEST full-featured MIDI interface available for the AMIGA 500/1000. Fully compatible with all programs that use serial MIDI standard output, ONLY the DELUXE MIDI INTERFACE offers you: SERIAL PORT Pass-thru, MIDI IN, MIDI THRU, and two SWITCHABLE THRU/OUTS! This is the optimum configuration for anyone with more than one synth or other MIDI device. No THRU on your synth(s)? No problem! Simply flip a switch on the Deluxe MIDI Interface, and avoid costly THRU boxes! Includes 8 foot cable, 1 year warranty.

Please Specify model 500/1000.....\$99

TTL Hi-res Monitor Adapter

For the Amiga 500/1000/2000. Plugs into RGBI port for ULTRACRISP flicker-free high resolution monochrome video output. Perfect for HI-RES GRAPHICS, WORDPROCESSING, CAD, DESKTOP PUBLISHING, BUSINESS, etc. ANY application in ANY resolution is SHARPER and CLEARER with the TTL Hi-res Monitor Adapter. Perfect for use with the low-cost (under \$100) Commodore 1601 monitor or Monitor 80. Includes easy installation instructions, a disk with a special Hi-Res WorkBench font and RGB port pass-thru (a 1080 RGB monitor may be used simultaneously). 1 year warranty.

BE KIND to your EYES!

(Monitor not included).....\$99.95

Light Pen

Designed to work with ANY Amiga program, the LIGHT PEN and DRIVER allows use of both your mouse and a pro-quality LIGHT PEN for the ultimate in precision graphics. Perfect for PAINTING, DRAWING, freehand SKETCHING, CAD and virtually ALL other AMIGA programs. Software features include: Single-pixel precision, variable sensitivity, ZOOM mode, button toggle, etc. Includes Inwell Systems' DT-184A high quality two button lightpen. **MAIL ORDER SPECIAL:** While supplies last, we'll also include a handy PEN-HOOK that attaches easily to your monitor to hold the AMIGA LIGHTPEN when not in use! (500/1000/2000).....\$129.95

All products are NOW SHIPPING! Please make cheque or money order payable to:

HyperTek/Silicon Springs

#120-1140 Austin Ave. Coquitlam, BC Canada V3K 3P5
Phone (604) 939-8235 Dealer Inquiries Invited.

All orders add \$4 shipping. All prices in US\$
Order by phone! VISA, Mastercard, AMEX welcome.
U.S. orders add \$3 handling fee.

AMIGA is a registered trademark of Commodore-Amiga, Inc.

- \ Now issue the two immediate commands:
- \
- \ MERGE "Text"
- \ MERGE "Fonts"
- \
- \ Now save the program before attempting to run it!

- \ Listing #6 - diskfont.bmap maker
- \ 'More Basic Text' by Bryan D. Catley
- \
- \ This program creates a diskfont.bmap
- \ file in the BasicDemos drawer of the
- \ current disk. The BasicDemos drawer must
- \ already exist.

```
PRINT "diskfont.bmap maker starting"
x$="OpenDiskFont"+CHR$(0)
x$=x$+CHR$(255)+CHR$(226)+CHR$(9)+CHR$(0)
x$=x$+"AvailFonts"+CHR$(0)
x$=x$+CHR$(255)+CHR$(220)+CHR$(9)+CHR$(1)+CHR$(2)+CHR$(0)
OPEN "BasicDemos/diskfont.bmap" FOR OUTPUT AS #1
PRINT #1,x$;
CLOSE #1
PRINT "diskfont.bmap maker ending"
END
```

•AC•

Two New products from Microbotics: The M501 memory expansion and the MultiFunction board

By John Foust

Microbotics has two new Amiga products—the M501 memory expansion for the Amiga 500 and a multi-function board for the Starboard II memory expansion.

The Amiga 500 has a small harbor on the underside. Commodore sells a memory expansion for this slot, model number A501. This board includes a battery-backed clock and an additional 512K of memory. It increases the total system memory of the Amiga 500 to one megabyte, equally divided between graphics (CHIP) and expansion (FAST) memory. (Programs are free to use CHIP memory, too.)

Several companies have produced memory expansions comparable to the Commodore A501 board. They compete against the Commodore board in terms of price. If you are buying an Amiga 500 system, you can save money by buying a third-party memory expansion. Third-party boards are designed to be identical to the Commodore board.

The M501, like the Commodore board, is shaped like a wedge of cheese. The harbor on the underside of the Amiga 500 is also this shape. The board rests on two rails inside the harbor.

Installing the M501 is fairly easy if you aren't scared by computer hardware. I feel comfortable opening a computer and mating a card edge connector, but the experience may be daunting for novices. I am sure any Amiga dealer will perform this installation if you are wary of doing it yourself.

Some difficulties are bound to crop up in mating the M501 with the Amiga. I had more trouble installing a Commodore A501 board than the Microbotics board, but this only proves there are manufacturing differences that cannot be avoided. Small differences in tolerance between all the connecting parts of the memory boards and the Amiga 500 require that some memory boards be inserted with different amounts of pressure.

The connector on the Amiga side is made up of two rows of fairly fragile metal pins that mate with a female connector on the M501. The memory expansion mates with tracks on both side of the harbor, and slides into position without much play. The manual is very simple—only a page of instructions. It recommends lining up the solder dots with the pins on the mating connector. This is good advice.

The same Okidata clock chip is used in the M501, the Commodore memory expansion, and on the Amiga 2000 motherboard. According to several engineers who have worked with this clock chip, it has some properties that may make it act irregularly in the future. The chip may be overly sensitive to the voltages supplied by the backup battery.

This problem, if it surfaces, will be endemic to all 100 percent compatible Amiga 500 internal harbor memory expansions (as well as the Amiga 2000). Only time will tell. Microbotics reports no unusual problems with the M501.

The M501 board has a list price of \$159.

Faster Floating Point

The Microbotics Starboard II is one of the most popular memory expansions for the Amiga 1000. An interesting part of its design is a connector on the main circuit board that can accept a daughter circuit board. This option allows for expansion within the expansion.

Microbotics planned at least two daughter boards. The first, a multi-function board, is now shipping; the second is a SCSI interface for adding SCSI devices, such as hard disks. According to Microbotics, the SCSI board should be out by the time you read this article.

The MultiFunction board has a battery-backed clock, a hardware RAM disk, parity checking, and a socket for a coprocessor chip that speeds up floating point mathematics.

To install the MultiFunction board, the Starboard must be removed from your Amiga, along with the cover. Depending on when your Starboard was made, you may be forced to change a chip on the board. The MultiFunction board comes with this replacement chip. Much like the installation of the M501 board, this procedure requires a certain fearlessness of hardware, as well as a small screwdriver. Installing the MultiFunction board presents a greater potential for damage (as

(continued)

APL.68000

\$99

A HIGHLY OPTIMIZED ASSEMBLER BASED APL INTERPRETER FOR FAST AND POWERFUL PROGRAMS. FEATURES A COMPLETE INTERFACE TO THE AMIGA ENVIRONMENT WITH PULL-DOWN MENUS, REQUESTER AND ALERT BOXES, SPEECH, SOUND AND GRAPHICS FACILITIES.

Order direct for \$99 + \$7 shipping, \$10 Canada.
VISA/MC/AMEX + 4% NJ res + 6% sales tax.

SPENCER
ORGANIZATION, INC.

P.O. Box 248 Westwood, N.J. 07675
(201) 666-6011

compared to the M501) because the memory chips and circuit boards are completely exposed during the operation. A little static electricity can lead to a several hundred dollar mistake.

The MultiFunction board comes with software to set the clock. Both CLI and Workbench versions are supplied. The CLI version can be added to your startup-sequence file to set the date each time your computer boots. The Workbench version is very simple to use.

Sticky Stuff

The Sticky Disk is a hardware-supported RAM disk. The entire two megabytes in the Starboard become a RAM disk that is not part of the Amiga's memory space, allowing the Sticky Disk to survive all crashes and warm reboots of the system. (With some software-based RAM disks, some software crashes disrupt the data in the RAM disk and all the data is lost.)

The Sticky Disk is attractive to software developers. Most programmers now use the recoverable RAM disk as part of their development tools. With it, they can compile, edit,

and test much faster than a floppy or hard disk based system. The entire Starboard must be devoted to the Sticky Disk; you cannot create a one megabyte Sticky and use the other megabyte for programs.

Sticky Memory is the name of the parity checking on the MultiFunction board. Parity checking is a way of detecting errors in memory chips. For every byte of memory, an extra bit is kept that remembers whether there was an odd or even number of bits in each byte. When this byte is read, the parity (odd or even) is checked. If the bit does not agree with the number of bits in the byte, then an error has been detected. To add parity checking to the Starboard, more memory chips (not supplied with the MultiFunction card) must be added.

Parity checking cannot detect all errors. If two bits change, an error may go undetected. There are also many other chips on a memory board; if these fail, the problem with the board will not be detected.

Parity checking is not error correction; it is only error detection. In other words, you get a warning—in this case, a recoverable Guru—that the memory is corrupt, and that you shouldn't trust any data in that memory. The IBM PC has parity checking. When an error is detected, the machine just hangs, with an error message on the screen.

Without parity checking, you would simply notice programs randomly crash with the memory board connected. Once this happened, you might determine that the memory board is malfunctioning and send it out for repair. I am not convinced of the ultimate utility of parity checking on the Amiga. Failing memory is a serious system problem which would certainly manifest soon enough.

The MultiFunction card also comes with its own version of software-based recoverable RAM disk, called JDDisk. This RAM disk can survive most

warm boots and crashes, but is not as robust as the Sticky Disk. It can use only a fixed portion of the Starboard memory space.

Floating Point

The MultiFunction board has a socket for a floating point coprocessor. This chip is manufactured by Motorola, the same company that made the 68000 microprocessor in your Amiga. A floating point chip increases the speed of floating point math operations in programs that know how to access the chip.

(Floating point operations are calculations involving real numbers. Most computer math operations are done with integer math—numbers with no fractional part after the decimal point. An example of a floating point number is 45.2315. Floating point math operations include functions like square roots and trigonometric functions like sines and cosines.)

In this design, the 68881 is treated as a peripheral device, not as a true coprocessor. The 68881 can only function as a true coprocessor when coupled with a 68020 microprocessor.

AmigaDOS 1.3 promises increased support for IEEE floating point libraries. The AmigaDOS 1.3 disk will have two IEEE floating point libraries: one in software like the present library and one that accesses a Microbotics-style 68881. A special file in the expansion drawer will tell the system which one to use.

You must install new floating point libraries on your Workbench to use the 68881. These libraries are supplied on the MultiFunction disk, along with a "read me" file that describes their installation. Microbotics has permission to distribute beta versions of the new libraries that will come with AmigaDOS 1.3.

How fast is the 68881? A SYSOP on the BIX computer network carried out

the Savage floating point benchmark using the Microbotics 68881 board. Willy Langeveld, moderator of amiga.user on BIX, showed the impressive increases in floating point operations possible with the Microbotics board.

Using the software implementation of the IEEE floating point format with a FORTRAN version of the Savage benchmark written in ABSOFT FORTRAN 2.2, it took 81 seconds to complete. (The C version was even slower.) Using the Starboard 68881 board and the IEEE library that uses this chip, with a version of Savage in Manx C version 3.4b, it took 1.57 seconds. If you write applications in C that use floating point, you can potentially access similar speed increases, depending on the program.

What about existing software? Currently, Microbotics is not aware of any commercial software that is enhanced by the 68881 board. Several software companies are reportedly preparing versions of their programs to work with the Microbotics 68881 card. So far, Microbotics has been contacted by Aegis Development, which is considering an enhanced version of Aegis Draw Plus; Byte by Byte, which is considering a new version of Sculpt 3D; and Electronic Arts, for an unknown product.

If you are interested in writing your own programs that use the 68881, the MultiFunction card comes with a disk of example code written in C and assembler. The examples and floating point libraries were written by Amiga wizard Dale Luck, Manx C author Jim Goodnow, and CLIMate co-author Dave Milligan.

The MultiFunction board lists for \$99.95. With the 68881 supplied by Microbotics, the board is \$379. Clearly, not many people need the 68881, so the MultiFunction board is mainly a (somewhat expensive) clock and an interesting alternative to the recoverable RAM disk.

✓ Check out our new price and features for Multi-Forth™
The Language of Innovation

Version 1.2 Multi-Forth increases the power, speed and flexibility of this already successful programming language and development tool. Some of the new features include:

- Local Multi-Tasking
- Sound Drivers
- Complete Set of Include Files
- New AmigaDos 1.2 Calls
- Enhanced Kernel

If you haven't tried Multi-Forth you may not have yet unleashed the full power of your Amiga.

Call our toll free number for a technical data sheet or check out our online services on CompuServe at **GO FORTH**.

Now only \$89.00

Creative Solutions, Inc.

4701 Randolph Rd, Suite 12 Rockville, MD 20852

301-984-0262 in MD

1-800-FORTH-OK (367-8465)

You can buy the MC68881RC-12A chip from an alternative source, such as an electronics supply house. A rough price would be in the \$250 range. Microbotics warns that some offers for low-cost 68881 chips may lead to trouble. Preliminary versions of the chip have a XC part number prefix. These are early, undebugged versions of the chip. Some companies are selling these at a discount price. The official version has an MC prefix. A 12 Mhz part is also necessary, so the part number should have a "-12" on the end.

At this time, it is hard to recommend the 68881 to most people. As someone once said about jazz, "If you have to ask, you don't understand it." It might similarly be said about the 68881, "If you have to ask what it is, you probably don't need it." At this time, it doesn't enhance any existing

software. If you are writing programs that use floating point, then you probably know how floating point coprocessors might help your application. At that point, your reason for buying a 68881 is a matter of weighing cost versus benefit.

Microbotics, Inc.
811 Alpha Drive Suite 335
Richardson, TX 75081
(214) 437-5330

M501 memory board \$159.
MultiFunction board \$99.95; with
68881 \$379.

•AC•

The Mindlight 7 and the People Meter

By John Foust

"Two unusual products for the Amiga"

You may have been first in your user group to break the two megabyte barrier on the Amiga. Or maybe you bought a 60 megabyte hard disk when everyone else had a 20 megger. But who has an Amiga color organ or an Amiga stress meter?

The Mindlight 7 and the People Meter are two unusual products for the Amiga. The Mindlight is an eighties version of the sixties color organ. The People Meter purportedly measures your level of relaxation or stress.

Mindlight 7

If you spent hours playing with the PolyScope program, you will like the Mindlight. An early page in the manual warns, "If you begin to get hypnotized, wait a little while before reading on."

The Mindlight itself is a round package about five inches across and an inch wide. The package is clear plastic, so you can see the circuit board inside. On one side is a DB-9 connector for connecting the Mindlight to the second mouse port. If you have anything plugged into the side of your Amiga, you can't install the Mindlight there. (At least one other hardware manufacturer fell into this trap, too.) If you don't have any extra memory installed, the Mindlight fits against the side of the Amiga, exposing its three thumbwheel controls to the topside of the Amiga.

When the Mindlight arrived, I went to Radio Shack and purchased a joystick extender cable for \$5. It works fine, and now I can use the Mindlight on my Amiga 500 as well. Visual Aural, the makers of the Mindlight, say they will soon have their own adapter for the Amiga 500. (This Radio Shack cable is also a very cheap source of pre-wired DB-9 connectors for making IBM AT serial cables or for replacing broken joysticks.)

The Mindlight hardware continuously and simultaneously measures five properties of the incoming sound: bass frequency, bass intensity, treble intensity, overall frequency, and overall intensity. The Mindlight software reads the incoming values and changes the screen display accordingly.

The Mindlight has a built-in microphone, as well as a 3.5 mm stereo mini-jack input. It accepts the level of audio signal commonly found on the "tape out" jack of a stereo system. This jack is normally an RCA phono type, so you need an adapter to connect your stereo directly to the Mindlight. The microphone is very sensitive. I did not find it necessary to connect it to a stereo. The microphone picked up the sound from the speakers quite well.

The three thumbwheel controls set the gain of the microphone, the overall gain, and something called "visual bias." Visual bias is the level at which

a sound triggers an on screen graphic. Essentially, it is a way to mask out ambient room noise.

Display Options

Dozens of options are available for controlling the display. So many, in fact, that only the most involved Mindlight owner could ever discover them all. The range of options makes the Mindlight more like an instrument; it is potentially very interactive. It is not a passive color organ that sits in the corner.

Capturing motion in words or even still screen shots is difficult. Once adjusted to the room, the Mindlight display is constantly moving. Stars appear and dance across the screen, leaving trails that slowly fade. Dancing triangles leave solid paths. Little objects that look like Sanskrit characters appear in starbursts. All this happens in real time and in synchrony with the sound.

The dozens of options include changing the resolution of the screen display to any mode supported by the Amiga (including HAM and overscan), changing the color palette and color cycle ranges, changing the way the images fade, loading and saving preset configurations, scrolling the screen in several directions, and scrolling through the individual color bitplanes.

One mode lets you use the mouse to draw on the screen while everything else is taking place. You can load IFF

pictures as backdrops, foregrounds, or the current screen itself. You can load IFF brushes and have the program move them around the screen in tune with the music. Another mode acts like a simple "pitch tracker." This mode lets you sing into the microphone, and the screen displays a chart that shows the musical note value of the current sound.

All these options obviously couldn't be included in menus. Some must be invoked with obscure keystroke combinations, such as "Enter Shift Up arrow."

I ran the Mindlight for several consecutive days on my Amiga 500. I set the controls for the microphone so it would do nothing if there was no sound in the room, but would be triggered by sounds as quiet as normal conversational levels. As I talked on the phone, I watched the Mindlight. If I turned on the stereo, the Mindlight "listened" to it, too.

I had several friends over during this time, and they all responded well to the Mindlight. We sat for minutes at a time, just interacting with the microphone and watching the screen. It is entertaining to see adults singing and making funny sounds just to get colorful patterns on the screen.

With a device like a color organ, you wonder how much of the display is random. Being curious about machines, I wanted to find the correlation between certain types of sound and the motion on the screen. At times, the correlation was very apparent. The objects on the screen were clearly responding to changes in frequency, and moving to one side or the other, based on the pitch of the current note. In other cases, the motion was tied to the amplitude of the sound.

I was surprised by the apparent intelligence of the software. It would sometimes lock into the beat of a song and synchronize its movements accordingly. As soon as the beat was

interrupted by a break in the song, the Mindlight stopped and radically changed its patterns.

The Mindlight would do very well in a dance club. So many dance clubs these days have projection screen televisions that flash music videos or clips from movies while the patrons dance. I think the Mindlight could be bundled with an Amiga 500 and sold to dance clubs and bars.

The Mindlight manual is just over one hundred pages long. The artwork on the disk and manual is in tune with the rest of the product—very spacey, very colorful. It is dedicated to people who "like to turn the music up and the TV down." It is quite thick, much thicker than you might imagine for such a product. It goes on and on, explaining modes with strange

*If you buy either
product, you'll
probably be the
first on your block
to have it.*

imaginary names such as Mozai and Kolai, and documenting all obscure modes and keystrokes.

The Mindlight software also supports MIDI instruments. I did not test this feature, but the description of MIDI recording and playback seemed a little crude. The manual also credits Golden Hawk Technology for help on the software. My guess is that the freely distributable MIDI code supplied with the Golden Hawk MIDI interface is incorporated into the Mindlight software.

Source code is included in C and assembly language for interfacing your own code to the Mindlight.

Background

The developers of the Mindlight, Mark Adams and Dan Egolf, say they have been working on visual music products for seven years. Their first efforts were stand-alone hardware devices. They then moved to the Commodore 64 and the Amiga. Adams is a scholar of Buckminster Fuller, and this influence shows through in the Mindlight and the TriClops game.

TriClops, a Visual Aural public domain game, was planned as a commercial product, but Adams was busy with the Mindlight. He is also developing a three-dimensional CAD program that may be sold through Visual Aural.

Mindlight Summary

The Mindlight 7 lists for \$148. If you are into gadgets and music, I think you'll like the Mindlight. It may not be your most practical purchase, but I can think of a few "real-world" uses (such as a video synthesizer in a dance club). You'll have to find your own justification for buying a Mindlight. It may be difficult to explain to your spouse or friends, but I think you will enjoy it just the same.

People Meter

A computer nerd, complete with lab coat, thick glasses, and pocket protector, is pictured on the cover of the People Meter package. His hair is orange and standing on end. He is wearing a blue bow tie, and on the ends of his fingers are two black bands with wires coming off them—this nerd is wearing Aminetics's People Meter.

The People Meter measures your stress level. You wear two sensors on your fingertips to measure the electrical resistance between the two fingers.

(continued)

ZING!Spell

Check and correct your spelling as you type!

\$ 79.95



P.O. Box 890408
Houston, TX. 77289-0408

(713) 488-2144

Credit Cards and Dealer
Inquiries Welcome!

This resistance is related to activity of your sweat glands. In some crude way, this resistance varies according to your emotional state. A polygraph (lie detector) usually measures sweat gland activity, along with many other bodily functions.

The People Meter hardware, which connects to the the second mouse port, is a small, white box with two knobs to control its sensitivity.

The People Meter comes with several programs on disk. One is called Moodbench. According to the manual, Moodbench was "inspired by the mood rings of the Seventies." (The manual tries desperately to be funny.) Moodbench changes the background color of your Workbench according to your stress level. As your stress level increases, the screen grows redder.

Another program is called Inter Action. While evaluating the People Meter, I double-clicked the icon for

this program and walked away while the program loaded. In another room of the house, I heard snoring. I thought to myself, "Wow, my neighbor must be snoring quite loudly, if I can hear it up here!" I walked around my apartment to find the source of the sound.

I found it in my computer room. On my Amiga 500 monitor screen was a digitized picture of a sleeping man. There were no fingers attached to the People Meter, so the program was snoring. Once fingers were connected to the People Meter, the pictures of the person on the screen flipped in quick succession between varying states of awareness. You can replace these digitized pictures with your own IFF pictures if you like.

Three other simple programs give bar graph or simple meter displays of your stress level. Example source code is included on disk for reading the People Meter settings.

A game called Stress 'N Bake is also included on the People Meter disk. I think it was inspired by that old episode of *I Love Lucy* where Lucy was hired at a cake factory, and the cakes start coming faster and faster. Your job is to run from floor to floor in the cake factory, carrying cakes from the conveyor belts to the truck on the ground floor. The mouse controls your movements. Mouse "play" is very fluid, making it difficult, if not downright frustrating, to get from floor to floor. Maybe this was done intentionally to increase your stress level. If your stress level increases, the conveyor belts move faster. If you advance to the second level, the cake factory turns into a Sidecar factory.

The premise of measuring stress by measuring electrical resistance between fingers is somewhat unbelievable. It is crude at best, at least in this configuration. The sensors are bands of cloth with an embedded metal screen that makes contact with your skin. It is far easier to change the stress measurement reading by moving your fingers than by relaxing. By flexing your fingers, you increase the amount of skin in contact with the metal mesh, changing the resistance. The manual warns against this unnecessary finger action and recommends that you don't move your People Meter hand. If you keep your hand steady, such as resting on the table, then the People Meter apparently does react to stressful events.

The People Meter was developed by Chris Irving, the hardware head behind the *Amazing Computing* internal memory upgrade (AC V2.1). The People Meter lists for \$59.95.

The Mindlight and the People Meter are both unusual devices. Their practical value may be hard to measure. One thing is certain, if you buy either one: You'll probably be the first on your block to have it.

•AC•

Winter Fantasy



SYMPHONY MUSIC VIDEO

SYMPHONY MUSIC VIDEO continuously displays pictures and plays music. At the end of a selection, another picture and music piece is loaded. The music and pictures are all in IFF format so you can modify ours with your favorite Paint or Music program or use your own music and pictures and create your own MUSIC VIDEO. You can even have your MIDI synthesizer play the music.

The MUSIC VIDEO that is included is perfect for the Christmas Season. Traditional Christmas scenes and music are continuously played. Fun to listen to and watch. Adds to the best of seasons. \$24.95

Symphony Songs

Ready to Play Music For

Deluxe Music, Music Studio, Sonix

A library of nearly 1,000 music masterpieces ready to play with your favorite music program. All selections are in both IFF and MUSIC STUDIO format. Space does not allow listing all songs in each volume, however, a few titles, the number of songs, and the total playing time is given.

Complete list of songs \$3.95.

Each volume is \$24.95 and includes the complete list.

BEATLES Part 1

Vol 15 (21 Pieces 40 Min)

Let It Be, Yesterday, Eleanor Rigby, When I'm 64, . . .

BEATLES Part 2

Vol 40 (15 Pieces 40 Min)

Magical Mystery Tour, Lucy In The Sky With Diamonds, Penny Lane, . . .

CLASSICAL Part 1

Vol 27 (18 Pieces 40 Min)

Prelude #1, Moonlight Sonata 1st and 2nd Movement, . . .

CLASSICAL Part 2

Vol 34 (13 Pieces 40 Min)

Sonata In C Major, Jesus Joy Of Man's Desire, . . .

CLASSICAL Part 3

Vol 31 (14 Pieces 35 Min)

1st Piano Concerto, Polonaise Sonata In C Major, Etude #3, . . .

CLASSICAL Part 4 (Bach)

Vol 35 (22 pieces 30 Min)

Two Part Invention #1, Three Part Invention #6, Prelude and Fugue 1, . . .

CLASSICAL Part 5 (Bach/Clementi)

Vol 46 (24 Pieces 50 Min)

Choral #1, Sonata #1, Theme and 11 Variations From The 2nd Sonata, . . .

BEETHOVEN, BROADWAY, & BLUES

Vol 38 (15 Pieces 40 Min)

2nd Movement Of the Pathetique Sonata, Minuet In G, Fuer Elise, . . .

COUNTRY CLASSICS Part 1

Vol 41 (15 Pieces 45 Min)

Thank God I'm a Country Boy, Act Naturally, . . .

ROCK Part 1

Vol 32 (19 Pieces 50 Min)

AXEL F, Eye Of The Tiger, Both Sides Now, . . .

ROCK Part 2

Vol 16 (20 Pieces 40 Min)

Georgy Girl, Guantanamera, Theme From "Love Story," Cherish, . . .

80's GREATEST

Vol 24 (15 Pieces 50 Min)

Hill Street Blues Theme, Chariots Of Fire Theme, Dynasty Theme, . . .

70's GREATEST

Vol 12 (19 Pieces 45 Min)

Tie A Yellow Ribbon On The Old Oak Tree, We've Only Just Begun, . . .

60's GREATEST

Vol 13 (19 Pieces 45 Min)

Windy, By The Time I Get To Phoenix, Come Saturday Morning, . . .

GOLD & PLATINUM HITS

Vol 45 (19 Pieces 60 Min)

Thriller, 99 Luft Balloons, California Girls, . . .

KENNY RODGERS HITS

Vol 39 (12 Pieces 45 Min)

Lady, Ruby, She Believes In Me, The Gambler, . . .

BILLY JOEL GREATEST HITS

Vol 43 (17 Pieces 65 Min)

Piano Man, Say Goodbye To Hollywood, Only The Good Die Young, . . .

COUNTRY CLASSICS Part 2

Vol 42 (13 Pieces 50 Min)

Ode To Billy Joe, Me and Bobby McGee, Country Roads, . . .

TV THEMES

Vol 37 (21 Pieces 35 Min)

Hill Street Blues, St. Elsewhere Theme, Masterpiece Theater Theme, . . .

MOVIE THEMES

Vol 19 (16 Pieces 40 Min)

MASH Theme, The Rose, Can You Read My Mind (Superman), . . .

BROADWAY'S THEMES

Vol 47 (25 Pieces 65 Min)

The Last Supper, Dr. Doolittle, The Old Dope Peddler, . . .

CHURCH MUSIC

Vol 23 (26 Piece 50 Min)

Amazing Grace, What A Friend We Have In Jesus, . . .

BARBERSHOP

Vol 22 (22 Pieces 45 Min)

Hello Dolly, Put On a Happy Face, Hey Look Me Over, . . .

RICHARD RODGERS SONGBOOK

Vol 18 (19 Pieces 40 Min)

Climb Every Mountain, DO-RE-MI, The Sound Of Music, . . .

NOSTALGIA

Vol 17 (22 Pieces 45 Min)

Let Me Call You Sweetheart, Ain't Misbehavin', On The Goodship Lollipop, . . .

CHRISTMAS

Vol 36 (24 pieces 50 Min)

O Little Town Of Bethlehem, Let It Snow, March Of The Toys, . . .

POLKA PARTY

Vol 33 (17 Pieces 40 Min)

Happy Polka, Pizzacato Polka, Betty Polka, . . .

We accept CASH, CHECK, COD, VISA and MASTER CARD orders.

Shipping and handling US and Canada . . . \$3.00

Shipping and handling outside the US and Canada . . . \$5.00

COD charge . . . \$2.00

Illinois residents add 6 1/4% sales tax.



Speech Systems

38W255 DEERPATH ROAD
BATAVIA, ILLINOIS 60510

(312) 879-6880

Deluxe Music Construction Set is a trademark of Electronic Arts.

Sonix is a trademark of Aegis.

Music Studio is a trademark of Activision.

The AMICUS Network

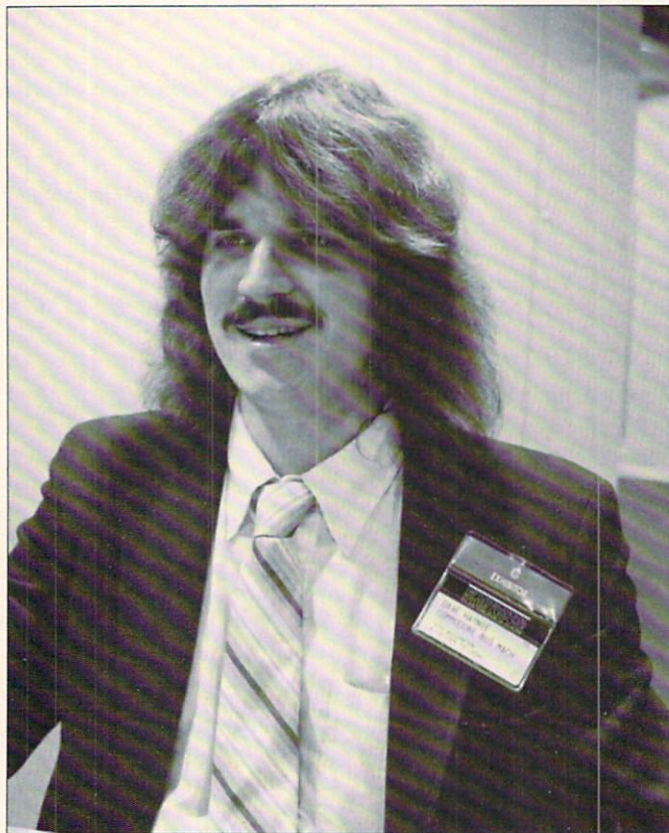
An Amiga Scrapbook

Something has been missing from all the stories I have brought back from Amiga shows: Pictures! Yes, there *are* faces behind the names.

An Amiga scrapbook would be wonderful for connecting faces to names like Carolyn Scheppner of Commodore technical support, Fred Fish of the disk collection, the SYSOPs of the national networks and bulletin boards, and the programmers behind popular programs.

Amigaphiles revel in trivia and personal details. Photographs can be great for personal stuff. Wouldn't you like to see Leo Schwab (of many display hacks) in his cape and hat? Joanne Dow (SYSOP of BIX) in her wizardess costume? David Joiner (of Microlusions and *Faery Tale Adventure*) in his alien insect costume?

There are other pictorial tales to tell. Would you like to see old advertisements of Amiga products, such as the original prototype "black box" Amigas, or the official Amiga joystick, or Suzy Chaffee on the Joyboard (a joystick you stood on)? How about the original proposals for the Amiga, which included an Apple II compatible disk drive and Apple BASIC in ROM?



Dave Haynie is an engineer at Commodore West Chester. He has been with Commodore for a long time, having worked on the design teams for the Commodore 64 and 128, and the Amiga 500 and 2000. He knows hardware and software, too. He is the author of DiskSalv and several articles for *Amazing Computing*. You'll also find him online, giving technical support on Usenet, Compuserve, BIX, and People Link.



Jeff Bruette was technical consultant for television's *Max Headroom* show, and was largely responsible for use of the Amiga on that short-lived series. Bruette continues to use the Amiga as a powerful, low-cost video computer in his work as a freelance video graphic artist in Hollywood.

Joy Weigel is the vice-president of marketing at Mimetics, the Amiga music company (makers of Sound-Scape). She and her partner, Bob Hoover, run Mimetics and present many road demonstrations of Amiga sound and graphics.

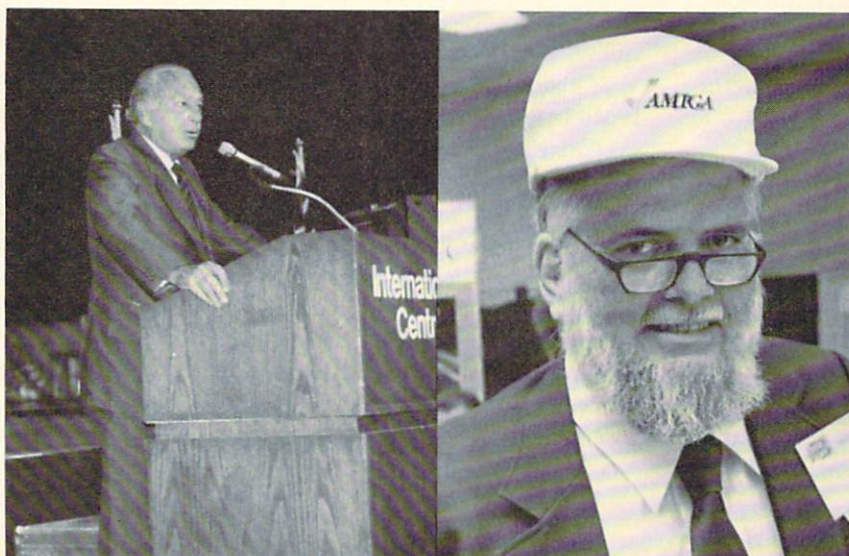


Scott Peterson is president of Byte by Byte, one-time makers of the PAL and PAL Jr. hard disk systems, and now makers of the Sculpt 3-D and Sculpt Animate modeling and animation programs.



Dale Luck is a full-time Amiga evangelist and survivor. He worked for Amiga Corporation long before Commodore came along, and still works there today. He designed large parts of the graphic software and its interface to the custom chips. Today, he heads the Commodore-Amiga West Coast office and is working hard on several projects, including the next revision of the operating system, auto-boot hard disks, X-Windows, and network interfaces.

Irving Gould is Chairman of the Board and majority stockholder of Commodore Business Machines. Gould helped start the company many years ago with partner Jack Tramiel. Here he is speaking at the World of Commodore show in Toronto.



Jay Miner is the principal mind behind the Amiga custom chip set. He came to Amiga Corporation from Atari, where he masterminded the chip at the heart of the Atari 800.

•AC•

P H A N T A S I E

by Kenneth E. Schaefer

For those who like nothing better than going out and killing a few hundred monsters.

It had to happen sooner or later. Someone has ported a game to the Amiga from the Atari ST. (Oops, I said the "A" word.) Phantasie is a fantasy role-playing game in the tradition of the Ultima series. Lovers of intricate plots and complex puzzles had best look elsewhere, but for those who like nothing better than going out and killing a few hundred monsters, Phantasie has a lot (of monsters) to offer. There are eleven cities, only 9 of which can be reached on foot (that's a subtle hint), ten dungeons, and three magic pools to discover and explore.

You start out in the town of Pelnor in the northeast corner of Gelnor. You create a party of six characters, very much like in Ultima III. The only characters that come on the disk are three elementals you will need later on in the game, so don't delete them! You choose each character's race, job, and name. The computer generates all your vital statistics, such as strength and intelligence, and asks if you want to keep this new character. "Yes" adds him to the guild roster; "no" deletes him. This process continues until you decide you have enough members in the guild. (You can have as many as 6 characters in one party, plus an elemental later on, so I recommend you create at least six initially.)

Next it's time to form a party of guild members. Choose your six, and

you're ready to leave the guild. Your next stop is the armory where you buy weapons. You have 1500 gold pieces to start with, since each character is given a nest egg of 250 when created. All gold is put into a common kitty, so you don't have to remember to share the wealth. Letting the fighters spend big bucks initially to get good weapons and armor is good strategy. Only a limited number of items is for sale initially anyway. Later, the magic users can spend some of your hard-won gold for the more expensive items. Of course, you'll be acquiring goods from your vanquished foes as well, so hardware won't be a problem for very long.

After buying equipment, save the party (via a menu selection) before starting off. If your party comes to an untimely demise, this step saves you from starting again from scratch. Phantasie comes with a "Back-up" utility, which lets you save and restore your characters to another disk. This touch is very reminiscent of Wizardry's separate utilities disk. Although Back-up is a separate program (you must exit Phantasie to use it; the game does not multi-task), it is a welcome feature. Back-up can only be accessed from CLI. The manual gives instructions on how to add CLI to your Phantasie disk, since it doesn't come on the disk (for some strange reason).

Now you're ready to explore and conquer. Why, there's a dungeon just outside Pelnor! That looks like a good place to start. You won't be in the dungeon very long, though, before you run into one of Phantasie's most frustrating shortcomings. Messages are often sent to you through windows opened just for that purpose. That's all well and good ... except that the length of time the text window remains open is controlled (seemingly randomly) by the program. It frequently isn't long enough to even read the message, let alone write it down (which you sometimes need to do). I tried every conceivable method (not moving the mouse, keeping the pointer within the window, setting speed at normal or even slow, putting no extra keyboard events in the buffer) to keep those messages from just blinking by, all to no avail. So keep on your eyes on the screen. Some important stuff is going to go screaming by.

While inside the dungeons, you get a good look at the combat system. It is very similar to Ultima III's system, with your party at the bottom of the screen and your foes at the top. Each foe even animates when he attacks. Some of the bobs for the monsters are highly original and funny. My favorite are the headless zombies; the baby dragons are a close second.

When combat begins, you are presented with a pop-up menu of your options, ranging all the way from "greeting" and "bribe" to "beg for mercy," with, of course, the ever-popular "attack" in there, too. You then must choose each character's actions. Before you write this task off as tedious, you may also opt to let a character repeat the last move he made. Each character and monster then takes his turn attacking, until one side is victorious.

If all this sounds fun, you're right. There are, however, several faults I must mention. Perhaps the most serious and certainly the most aggravating fault is the quick-close text windows. A not-so-serious complaint is that Phantasie appears to be a "straight" port, with no attempt made to enhance the Atari version when they brought it over to the Amiga. The game has only 16 colors, no music during play (the intro song is so strange they could have left it out), no secondary menus, and no multi-tasking. In short, Phantasie looks and plays just like your Amiga had been transformed (some might say crippled) into an ST. This isn't too much of a detraction, but the time it took for this game to reach the Amiga makes me wonder why there are no improvements.

The graphics are certainly better than, say, Ultima III, but not as good as Bard's Tale. The combat system is more elaborate, but the spell list is smaller and simpler. The user interface combines mouse input and pop-up menu selection, giving the game an uneven, back-and-forth feel. The only way to exit the game is to reboot your machine. With all these complaints (most of them nit-picky, I admit), why did I like playing the game, and keep at it until I won? Maybe I'm helplessly addicted to "hack and slash" games. Maybe Phantasie's interesting story kept me going; I was always curious to read the next scroll (which reveals the history of Gelnor and its major landmarks). Maybe it's just very playable. Whatever the reason, despite its shortcomings, I enjoyed playing Phantasie.

Those of you with more than 512K (especially you Amiga 500 and 2000 users new to our ranks ... while I'm thinking about it, welcome!) will have problems playing Phantasie. The bobs are scattered and the music doesn't play. This type of "not-playing-by-the-rules" is really disappointing from a company as big as SSI. But, take heart! Type in the short text file (using Ed or some other pure ASCII editor) below, copy it to ram:, and execute it from there. (It will also work if you have a second drive.) This addition will allow you to play Phantasie with your extended memory. (Make sure that nofastmem is in your system drawer!)

```
run system/nofastmem
assign fonts: Phant:fonts
stack 4000
cd Phant:
game
```

That's all there is to it.

As I intimated before, Phantasie lacks some of the "polish" we have become accustomed to, but it is still enjoyable and playable. Phantasie III is on its way (whatever happened to Phantasie II?) and I certainly expect all the bugs to be worked out. In the meantime, those who love making a fantasy world a better place by killing thousands of monsters have another game to add to their ever-growing collections.

Now, in the continuing tradition of Amazing Gaming Utilities, I present the Amazing Phantasie Character Editor. Although it is a little long, it allows you to create characters at any level of "lethality." If you just want to modify your characters so they have some extra hit points, you can do that. If you want to create characters that need fear nothing save Zeus himself, you can do that, too. (By the way, there is no way to defeat Zeus, even if you wanted to, which you don't—another free hint.)

```
CLEAR ,10000
CLEAR ,80000

REM Amazing Phantasie Character Editor
REM V1.0 by Kenneth (DirkMaster) Schaefer

DIM SHARED shields$(20), armors$(20), Weapons$(60), weaponreq(60)
DIM SHARED armorreq(20), spellist$(54), guild$(25), finals$(7050)

LET start = 1
LET location = 0
LET again = 0

FOR w = start TO 20
  READ shields$(w)
NEXT w

FOR x = start TO 20
  READ armors$(x)
NEXT x

FOR y = start TO 60
  READ Weapons$(y)
NEXT y

FOR z = start TO 60
  READ weaponreq(z)
NEXT z

FOR v = start TO 20
  READ armorreq(v)
NEXT v

FOR u = start TO 54
  READ spellist$(u)
NEXT u

REM Open file on disk where characters live
  and read into memory
OPEN "Phant:guild.dat" FOR INPUT AS #1
FOR j = start TO 25 ' slots for 25 characters,
  including elementals
  guild$(j) = INPUT$(282,#1) 'each slot is 282 bytes long
NEXT j
CLOSE #1
PRINT "Welcome to the Amazing Phantasie Character Editor!"
```

(continued)


```

WHILE again = 0
  PRINT "Please enter the name of the character
  you want to edit."
  PRINT "or press <return> to exit"
  PRINT
  LINE INPUT itsme$
  LET uname$ = UCASE$(itsme$)

  LET location = start      ' find that character
                             in one of the arrays
  LET check$ = RIGHT$(guild$(location),12)

  WHILE LEFT$(check$,LEN(uname$)) <> uname$ AND location < 25
    LET location = location + 1
    LET check$ = RIGHT$(guild$(location),12)
  WEND

  IF itsme$ = "" THEN      ' no name input, must want to quit
    PRINT "okay. If that's the way you want to be!"
    LET again = 1
  ELSEIF location > 0 THEN  ' we found the dude
    PRINT "I found him! He's the "; location:
    PRINT "character in your file!"
    CALL Worker (location)
  ELSE                      ' must not have found him
    PRINT "Huh? Who? I'm confused.
    Never heard of 'em. Try again."
  END IF
WEND

CALL CLOSEUPSHOP (guild$)

'Shields
DATA Glove, Wooden Shield, Wdn Shield +1, Small Shield,
  Small Shield +1
DATA Small Shield +2, Small Shield +3, Medium Shield,
  Med. Shield +1
DATA Med. Shield +2, Med. Shield +3, Large Shield,
  Lrg Shield +1
DATA Lrg Shield +2, Lrg Shield +3, Giant Shield, Gnt Shield +1
DATA Gnt Shield +2, Gnt Shield +3, God Shield

'Armor
DATA Clothing, Robes, Leather, Hard Leather, Ring Mail,
  Scale Mail, Chain Mail
DATA Splint Mail, Banded Mail, Plate Mail, Cloth +1,
  Robes +1, Leather +1
DATA Leather +2, Ring Mail +1, Ring Mail +2,
  Chain Mail +1, Chain Mail +2
DATA God Robes, God armor

'Weapons
DATA Stick, Knife, Small Club, Small staff, Small mace,
  Dagger, Small flail
DATA Club, Mace, Small Hammer, Small axe, Staff,
  Short sword, Flail, Hammer
DATA Pitch fork, Spear, Axe, Sword, Heavy mace, Maul,
  Trident, Large Spear
DATA Large axe, Morning Star, Pike, Long Sword, Spetum,
  Bardiche, Halbred
DATA Small mace +1, Dagger +1, Small mace +2, Dagger +2,
  Dagger +3, Staff +1
DATA Dagger +4, Flail +1, Spear +1, Axe +1, Sword +1,
  Sword +2, Sword +3
DATA Large axe +1, Sword +4, Sword +5, Sword +6,
  Halbred +1, Sword +7, Halbred +2
DATA Halbred +3, Sword +10, Halbred +4, Halbred +5,
  Halbred +6, Halbred +7
DATA God knife, God mace, God axe, God sword

'Weapons - required stat's (Strength * 2 + Dexterity)
DATA 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28,
  30, 32, 34, 36, 38
DATA 40, 42, 44, 46, 48, 50, 52, 54, 56, 58, 60, 2, 4,
  6, 8, 10, 12, 14, 16
DATA 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38, 40, 42,
  44, 46, 48, 50, 52
DATA 54, 56, 58, 60

'Armor - required stat (strength)
DATA 2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 0, 0, 2, 4, 6,
  8, 10, 12, 14, 16

'Spells
DATA Healing 1, Healing 2, Healing 3, Healing 4,
  FireFlash 1, FireFlash 2
DATA FireFlash 3, FireFlash 4, Quickness 1, Quickness 2,
  Quickness 3, Quickness 4
DATA Strength 1, Strength 2, Strength 3, Strength 4,
  Protection 1, Protection 2
DATA Protection 3, Protection 4, Confusion 1, Confusion 2,
  Confusion 3

```

```

DATA Confusion 4, Weakness 1, Weakness 2, Weakness 3,
  Weakness 4, Binding 1
DATA Binding 2, Binding 3, Binding 4, MindBlast 1,
  MindBlast 2, MindBlast 3
DATA MindBlast 4, FlameBolt 1, FlameBolt 2,
  FlameBolt 3, FlameBolt 1, Charm
DATA Sleep, Teleportation, Resurrection, Ninja 2, Fear, Dissolve
DATA Sum. Elemental, Dispell Undead, Ninja 1, Awaken,
  Monster Eval.
DATA Vision, Transportation
END

SUB Worker(location) STATIC
LET start = 1

  Strength:
  PRINT
  PRINT "Please input new strength"
  INPUT "values between 1 and 255 ", stren
  IF stren > 255 OR stren < 1 THEN GOTO Strength
  LET test$=CHR$(0)+CHR$(stren)
  MID$(guild$(location),1,2)=test$

  Intelligence:
  PRINT
  INPUT "Now enter new Intelligence ", intell
  IF intell >255 OR intell < 1 THEN GOTO Intelligence
  LET test$=CHR$(0)+CHR$(intell)
  MID$(guild$(location),3,2)=test$

  Dexterity:
  PRINT
  INPUT "Now for the Dexterity ", Dex
  IF Dex > 255 OR Dex < 1 THEN GOTO Dexterity
  LET test$=CHR$(0)+CHR$(Dex)
  MID$(guild$(location),5,2)=test$

  Constitution:
  PRINT
  INPUT "next is constitution ", con
  IF con > 255 OR con < 1 THEN GOTO Constitution
  LET test$=CHR$(0)+CHR$(con)
  MID$(guild$(location),7,2)=test$

  Charisma:
  PRINT
  INPUT "now for charisma ", char
  IF char > 255 OR char < 1 THEN GOTO Charisma
  LET test$=CHR$(0)+CHR$(char)
  MID$(guild$(location),9,2)=test$

  Luck:
  PRINT
  INPUT "and finally luck ", luc
  IF luc > 255 OR luc < 1 THEN GOTO Luck
  LET test$=CHR$(0)+CHR$(luc)
  MID$(guild$(location),11,2)=test$

  PRINT
  PRINT "Now for something completely different!"
  PRINT "Magic points and Hit points,
  with a maximum of 255 each"

  MagicPoints:
  PRINT
  INPUT "Input magic points ", mp
  IF mp > 255 OR mp < 1 THEN GOTO MagicPoints
  LET test$=CHR$(0)+CHR$(mp)
  MID$(guild$(location),17,2)=test$
  MID$(guild$(location),19,2)=test$

  HitPoints:
  PRINT
  INPUT "and how about those hit points ", hp
  IF hp > 255 OR hp < 1 THEN GOTO HitPoints
  LET test$=CHR$(0)+CHR$(hp)
  MID$(guild$(location),21,2)=test$
  MID$(guild$(location),23,2)=test$

  PRINT
  PRINT "Now that we have all that out of the way..."
  PRINT "How about setting those attributes?"
  PRINT "These values can also go to 255"

  Attack:
  PRINT
  INPUT "First there is attack ", att
  IF att > 255 OR att < 1 THEN GOTO Attack
  LET test$=CHR$(0)+CHR$(att)
  MID$(guild$(location),31,2)=test$

```



```

Spot:
PRINT
INPUT "next is spot items ", sp
IF sp > 255 OR sp < 1 THEN GOTO Spot
LET test$=CHR$(0)+CHR$(sp)
MID$(guild$(location),37,2)=test$

Parry:
PRINT
INPUT "now for the parry ability ", par
IF par > 255 OR par < 1 THEN GOTO Parry
LET test$=CHR$(0)+CHR$(par)
MID$(guild$(location),33,2)=test$

Disarm:
PRINT
INPUT "next is Disarm Traps ", dis
IF dis > 255 OR dis < 1 THEN GOTO Disarm
LET test$=CHR$(0)+CHR$(dis)
MID$(guild$(location),39,2)=test$

FindItem:
PRINT
INPUT "now for Find Item ", find
IF find > 255 OR find < 1 THEN GOTO FindItem
LET test$=CHR$(0)+CHR$(find)
MID$(guild$(location),35,2)=test$

PickLock:
PRINT
INPUT "how about that Pick Lock ", pick
IF pick > 255 OR pick < 1 THEN GOTO PickLock
LET test$=CHR$(0)+CHR$(pick)
MID$(guild$(location),45,2)=test$

Listen:
PRINT
INPUT "next is the all important Listen ", lis
IF lis > 255 OR lis < 1 THEN GOTO Listen
LET test$=CHR$(0)+CHR$(lis)
MID$(guild$(location),41,2)=test$

Swim:
PRINT
INPUT "finally, there is Swim ", sw
IF sw > 255 OR sw < 1 THEN GOTO Swim
LET test$=CHR$(0)+CHR$(sw)
MID$(guild$(location),49,2)=test$

PRINT "Gold in bank has a ceiling of 65535"
GIB:
PRINT
INPUT "How much gold do you have in the bank ", gold
IF gold > 65535 OR gold < 1 THEN GOTO GIB
LET high.byte=INT(gold/256)
LET low.byte=gold-(high.byte*256)
LET test$=CHR$(high.byte)+CHR$(low.byte)
MID$(guild$(location),260,4)=test$

PRINT "Now, how about that Experience level?"

EXPERIENCE:
PRINT
PRINT "I'm keeping it within reason
so that you can still have fun"
INPUT "what is your experience level ", exper
IF exper > 1000000 OR exper < 1 THEN GOTO EXPERIENCE
LET high.byte = INT(exper/65535)
LET mid.byte = INT((exper-(high.byte*65535))/256)
LET low.byte = exper-((high.byte*65535)+(mid.byte*256))
LET test$ = CHR$(0) + CHR$(high.byte) +
CHR$(mid.byte) + CHR$(low.byte)
MID$(guild$(location),267,8)=test$

WIDTH 76, 19
PRINT
PRINT "Okay, now for some really fun stuff!"

PickWeapon:
PRINT
PRINT "Choose a weapon and enter its letter"
PRINT "If you can't use it,
you'll be told and allowed to pick another"
FOR w = start TO 60
PRINT w ;Weapons$(w),
NEXT w
PRINT
INPUT "which is your weapon of choice ", weap
IF weap > 60 OR weap < 1 THEN
PRINT "just pick from what is listed above, please!"
GOTO PickWeapon
END IF
LET req = ((stren * 2) + Dex)
IF weaponreq(weap) > req THEN
PRINT "I'm afraid you're not
talented enough for that one"

```

```

GOTO PickWeapon
END IF
LET test$ = CHR$(0) + CHR$(weap + 40)
MID$(guild$(location),229,2) = test$
MID$(guild$(location),87,2) = test$

PickArmor:
PRINT
PRINT "Now how about some armor?"
FOR x = start TO 20
PRINT x ;armors$(x),
NEXT x
PRINT
INPUT "which piece of armor do you want ", armr
IF stren < armorreq THEN
PRINT "sorry, but you're not strong
enough to use that"
GOTO PickArmor
END IF
IF armr > 20 OR armr < 1 THEN
PRINT "you can only pick from 1 to 20"
GOTO PickArmor
END IF
LET test$ = CHR$(0) + CHR$(armr + 20)
MID$(guild$(location),231,2) = test$
MID$(guild$(location),85,2) = test$

PickShield:
PRINT
PRINT "and finally, how about a shield?"
FOR y = start TO 20
PRINT y;shields$(y),
NEXT y
PRINT
INPUT "please pick a shield ", shld
IF shld > stren THEN
PRINT "sorry, you're not strong enough
to carry that shield"
GOTO PickShield
END IF
IF shld > 20 OR shld < 1 THEN
PRINT "Only what ya see, buddy!"
GOTO PickShield
END IF
LET test$ = CHR$(0) + CHR$(shld)
MID$(guild$(location),233,2) = test$
MID$(guild$(location),83,2) = test$

MagicSelect:
LET again = 0
PRINT
PRINT "Lastly, but not leastly,
we got dem wonerful spells!"
PRINT "If there are any limits here,
I haven't found 'em!"
PRINT "Have fun! Don't make it boring."
WHILE again = 0
FOR u = start TO 54
PRINT u;spellist$(u),
NEXT u
PRINT
INPUT "Which one do you want,
<return> to end "; newspell
PRINT
IF newspell = 0 THEN
LET again = 1
END IF
IF newspell > 54 THEN GOTO MagicSelect
LET test$ = CHR$(0) + CHR$(start)
MID$(guild$(location),((newspell * 2) + 95),2) = test$
WEND
END SUB

SUB CLOSEUPSHOP (guild$) STATIC
OPEN "Phant:guild.dat" FOR OUTPUT AS #1
FOR j = start TO 25
final$ = final$ + guild$(j)
NEXT j
IF LEN(final$) <> 7050 THEN
PRINT "Whoops! Something's wrong here!"
PRINT "File not the right size.
Aborting write to disk"
CLOSE #1
STOP
ELSE
PRINT "Thanks for using the Amazing Phantasie
Character Editor!"
PRINT# 1,final$,
CLOSE #1
END IF
END SUB

```

•AC•

Roomers

By the Bandito

Lasers, Viruses, Sacs and Wedges

The arrival of the Amiga 3000 was announced on the national television show *Computer Chronicles*. The specifications were truly amazing, the reporter said—but everything described was completely possible with an Amiga 2000.

So much for rumors and the quality of the news on *Computer Chronicles*. It sounds like *Computer Chronicles* gets its information from Compuserve's *Online Today*, an electronic newspaper. *Online Today* reported a similar rumor about the Amiga 3000. Several months ago, *Online Today* also reported that Commodore recalled the operating system to the Amiga 1000. What? You don't remember that? Neither does the Bandito.

The Bandito would like to admit that the recent thread of rumors about the Laser Toaster has nothing to do with the NewTek Video Toaster. NewTek did not steal the "Toaster" name from this column. Indeed, the opposite is true; "Laser Toaster" was the NewTek code word for the Video Toaster. The Laser Toaster became a tremendously popular subject of conversation for cocktail parties at Amiga shows, so the Bandito decided to share the joke with you.

However, this doesn't mean that the concept of a Laser Toaster wasn't taken seriously by some people. I swear on a stack of *National Enquirers*, there is a company in the Northwest

developing a laser toaster as described in this column. The Bandito has it on good authority from experts in the field of lasers that such a thing is possible, and even profitable.

In place of the Toaster rumor, a new rumor has been started about Digi-Word, a HAM word processor, also from NewTek. According to this rumor, NewTek used Digi-Word to lay out the ads for the Video Toaster. You can have any of 4096 colors for any character in your message. Then the rumor-spreader started saying something about animation on the printed page, but the Bandito walked away at that point...

ScrOnAlyze!!!.c

A contract programmer for Micro Systems Software reveals that Scribble!, Online!, and Analyze! are indeed the same program, called ScrOnAlyze!!!. (This is pronounced "skron-a-lies.") The source code to these programs is actually one large C file named "ScrOnAlyze!!!.c." The executable programs are conditionally compiled based on such preprocessor directives as "#ifdef ONLINE" and "#ifdef ANALYZE." A spokesman would neither confirm nor deny the story. (This is a programmer joke, not a real rumor, so if you don't get it, just read the other rumors. Trust me, it was a funny joke. If you want to look like a hotshot C programmer at your next user group meeting, just memorize this paragraph and repeat it verbatim.)

Commodore Amiga in Los Gatos may be re-hiring even more of the old technical team in months to come. Commodore has even placed employment advertisements in electronics magazines, asking for people skilled in operating systems and video graphics. Neil Katin is working on a project, and Jim Mackraz recently left Mitsumi for a chair in the Amiga offices. Recent postings on Usenet indicate they are courting more and more former Los Gatos people to re-join the staff to further enhance the Amiga hardware and software.

Other higher-resolution displays will be possible in the Amiga 2000, according to insiders. This can be done by simply by replacing the "graphics.library," which does not require replacing the Kickstart ROM. The new displays may come from third-party developers, not Commodore. Possible graphics hardware includes the TI 34010 chip, a commercially available graphics coprocessor that is about as powerful as the 68000 itself, with higher resolutions than the Amiga chipset. All Amiga software should still work correctly, if the programmer doesn't resort to tricks that circumvent the operating system function calls. However, the output may not be NTSC compatible.

At a recent user group meeting, an Amiga staffer dropped hints about the release of a new Amiga machine in mid-summer, as well as progress on

AmigaDOS 1.4. One improvement would be information in ".info" files so that icons wouldn't be misproportioned on interlaced or other display sizes. They may try to get the Greenhills C compiler running native on the Amiga, a step toward their long-standing goal of native-compiling the operating system code on Amiga 2000 systems, freeing them from the Sun systems now needed for the task.

Commodore insiders say the 80286 AT-style Bridge Card will be ready soon, and users should expect it to replace the 8088 BridgeBoard. Although the 80286 processor will run programs faster than the 8088 BridgeBoard, the screen updates will be just as slow. The Amiga 2000 must run a program to display the PC screen in a window on the Amiga screen, and the transfer process can't keep up with the faster card, so screen scrolling will not be as smooth as a regular PC monitor. Users are dreaming of a 80386 card, too. Some rumors say it is in development in West Germany, but cynics say the PC bus speed in the Amiga 2000 is limited, so the 80386 will not be used to the fullest.

Amiga on TV

The Disney Channel showed previews for a movie called *Not Quite Human*, starring Alan Thicke. The movie will include the Amiga as part of the plot, and Amiga screen graphics will be used in the show. The computer has a starring role! The Amiga was also used in a recent HBO movie, *Into the Homeland*.

The Amiga 500 made an appearance on the NBC *Today* show, on a segment about "adult toys." The Amiga was presented as a home video computer system. Newscaster Jane Pauley was shown writing holiday greetings in color-cycled animation.

Sacs and Wedges

The "Magic Sac" Macintosh emulator remains far from market. According to a source close to the company, no one is working on it at this time. The task has shifted from programmer to programmer, and now the company thinks it has higher priorities.

Look for a device called the Wedge to come from Canada. This card for the Amiga 2000 lets the Amiga side access a standard IBM-type hard disk controller without the presence of the BridgeBoard. The price is around \$150, and IBM-type hard disks and XT controllers are almost a dime a dozen. A Wedge owner could add a hard disk to an Amiga 2000 system for less than \$500. Call Dave Allen at (604) 270-0064 for more information.

A large West coast Amiga dealer pulled a fast one on Commodore and angered many other dealers recently by ordering a truckload of Amiga 2000s and peripherals, insisting there were standing orders for them all. (The Commodore rep for the area was new and didn't know he had to investigate the claim.) The dealer held a parking lot sale at near-list prices to captive, drooling Amigaphiles who had been waiting weeks because of a lack of hardware on the West coast. Other dealers were incensed, seeing that they got the scraps of the sale. The dealer sold only a fraction of the hardware in the truck, and Commodore had to store them in the local warehouse.

Word Perfect 5.0

Word Perfect version 5.0 for the Amiga is at least two years away, according to a source close to the company. The program will use its own fonts and be available for other computers as well. The Amiga is a test bed for much of the development. Word Perfect Library and Notepad are in beta test; they appeared at a recent computer show and in the hands of beta testers.

Electronic Arts' flight simulator game *Interceptor* may be out now, or may be tied up in litigation, depending on who you talk to. A super taste of it has been wowing the demo disk circuit. A real copy of it was stolen by pirates at the World of Commodore show. It appeared on the pirate bulletin boards that afternoon. Aegis also lost a copy of VideoTitrer, which had not shipped at that time.

Digital Creations' "D' Buddy" HAM editor program was bought by Electronic Arts. It will now be called Deluxe Photo Lab. Some features were cut and others were added, to make it an EA product. Don't expect to see it until March.

Progressive Peripherals has revived TeleCraft, a telecom program once planned for production by Commodore.

Aegis Goes Mac II

Aegis Development is shifting gears to develop products for the Mac II, after showing very preliminary versions of Mac II animation and presentation graphics programs in a private suite at a recent computer show. Sources say Leo Schwab's "The Dream Goes Berserk" animation was ported to the Mac II, and that it was playing in the Apple booth.

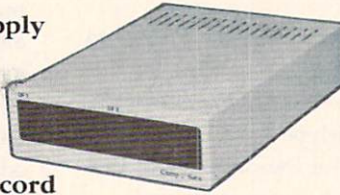
According to one rumor, an Apple spokesman asked Aegis to replace the Amiga ball with an apple. However, this was not possible, as only the data of the animation was ported to the Mac II, not the VideoScape program itself. As Commodore's Andy Finkel once said, "Any sufficiently advanced technology is indistinguishable from a rigged demo."

Sure, Leo's animation can play on the Mac II, but can they slide the screen down to reveal another application running in the background?

AMIGA DUAL 3 1/2" DISK DRIVES

100% Compatible with
Amiga 500, 1000 & 2000 Computers

- Internal Power Supply
- All Metal Chassis
- Horizontal Layout
- Vented Enclosure
- On-Off Switch
- 6 foot 3-prong linecord
- Primary Circuit Breaker Protected
- Color Coordinated to Amiga Computers



ONLY \$395.00

20 Meg Hard Drive (SCSI) with Controller

ONLY \$785.00

Now Shipping Single-Disk Drive Units

Comp-U-Save

414 Maple Avenue, Westbury, NY 11590

In NY State (516) 997-6707

Outside NY State (800) 356-9997

"Meeting the Needs of People in the Electronic Age"

Commodore is preparing to launch a PC clone line featuring a machine tentatively named Colt. (The PR agency doesn't like the name, but Commodore officials do.) This machine is dual-speed XT-class, with 640K of memory, a 3 1/2 inch disk drive, CGA color graphics, and a color monitor.

Two other models in the line are still unnamed. One is another XT-type Colt with a one-third faster clock speed and maybe a hard disk. The third machine is an 80286-based AT-class computer. These are boxed and ready to go. Commodore hoped to launch them for the holiday selling season, but missed the mark.

The NYIT digitizer board once took three slots in the Amiga 2000, but is now back in the shop for a redesign that would make it just one or two boards again. Because of this set-back, availability is still unknown.

The COMAL language is now in beta testing for the Amiga. "COMAL-ites" have waited a long time for this one to come over from Europe. There's no word yet on when it will be available, but it should be here soon.

•AC•

Virus News

The Swiss pirate programmer who created the disk virus sent a letter to Commodore technical support, explaining why he wrote it and offering a utility as a peace offering. The program detects and eliminates the virus. He boldly suggested that the program be included on a Workbench disk. The pirate said he didn't write the virus to destroy software, but to prove a point to a fellow programmer. The virus came to America on a disk of pirated software.

The virus code in the root block of a disk keeps a count of how many times it has duplicated itself. The detector program will print this number for you. It will also remove the virus from memory or erase it from a disk, and protect a disk from being infected. It is not known if the protector program

has another virus in it. Close examination of the code shows that parts of the program are encrypted data, meaning the program itself may be infected with a different virus. Don't try this at home, kids.

At press time, there came word of a second type of disk virus that came to the United States on a game called Roadwar Europa.

Commodore is planning to do more television promotion of the Amiga 500. There is even talk of a touring national spokesperson who would appear on news programs (similar to the recent *Today* show appearance). The Amiga might even make it to *Late Night With David Letterman*!

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ cannot be held responsible for the reports made in this column.]

By Gerald Hull

Part I of this article (AC V2.12) discussed the origins of LIFE in cellular automata theory. We touched on its rules, its history on the Amiga, and some of its curious properties and implications. In Part II (AC V3.1), we took a look at efficient use of the Amiga blitter. In Part III, we conclude with the nine blit calculation of a LIFE generation, and the source to LIFER.

Talkin 'bout My Generation

The algorithm I used is essentially identical to the one Scott Evernden borrowed from the Mark Niemiec article, "Life Algorithms" (BYTE, January 1979). First, you sum a cell's neighbors in each horizontal row (don't include the cell itself in these calculations!), and then sum those sums in turn. If there are exactly three neighbors, or a living cell has two neighbors, a living cell is in the next generation.

		2's bit	1's bit		
		\	/		
0 1 1	=		1 0		x x x
0 1 0	=		0 0		x 1 x
1 0 0	=		0 1		x x x

			1 1		

->

By using the ASHIFT and BSHIFT blitter registers, you can operate on three horizontally adjacent cells with the same blit. Suppose we have the simplest case: one pixel (or bit) per cell. We want to shift the left cell one bit to the right, the right cell one bit to the left, and leave the center cell where it is. Address those cells with the A, B, and C bitplane pointers, respectively.

The ASHIFT value is easy: 1. What about BSHIFT? To shift one bit left with the blitter, you need to shift it $16 - 1 = 15$ bits to the right, and then address it through the succeeding word, yielding the desired result ($= 15 - 16 = -1$). At this point, you might think about using BLTAFWM and BLTALWM to avoid accidentally destroying valid data. However, the wraparound border guarantees these shifts have no "undocumented features."

Using these sorts of shifts, we can get the horizontal neighbor sums in just four blits. The first and third rows can be "doubled" together with an extra tall image, saving two blits. (This is the trick I overlooked.) How do we sum

(continued)

Table One
A Nine Blit Calculation of a LIFE Generation

BITPLANES					MINTERMS							
D	A	B	C		C	-C	C	-C	C	-C	C	-C
					B	B	-B	-B	B	B	-B	-B
					A	A	A	A	-A	-A	-A	-A
(1)	R1	a/f	b/g	c/h	1	0	0	1	0	1	1	0
(2)	R2	d	e		0	0	1	1	1	1	0	0
(3)	R4	R1/	R2	/R1	1	0	0	1	0	1	1	0
(4)	R5	R1/	R2	/R1	1	1	1	0	1	0	0	0
(5)	R1	a/f	b/g	c/h	1	1	1	0	1	0	0	0
(6)	R2	d	e	R5	0	1	1	0	1	0	1	0
(7)	R3	R1/	R2	/R1	0	0	0	1	0	1	1	0
(8)	R1	R5	R3	R2	1	0	0	0	1	1	0	0
(9)	screenZ		R1	R4	1	1	0	0	1	0	0	0

Remarks

I use "R1" to "R5" to represent the "register" bitplanes, and a slashbar for clarity when "doubled" bitplanes are involved. The cell and its neighbors are addressed as the diagram indicates:

a b c d e f g h

The numbered blits represent the following calculations:

- (1) 1st and 3rd row 1's bit sum
- (2) 2nd row 1's bit sum
- (3) total 1's bit sum
- (4) total 1's bit carry
- (5) 1st and 3rd row 2's bit sum
- (6) 2nd row 2's bit sum plus total 1's bit carry
- (7) 1st and 3rd row 2's bit carry
- (8) total 2's bit sum with no further carry
- (9) a new generation

these in turn? The 1's bit (LSB) sum for the entire neighborhood is easy, but to get the 2's bit (LSB+1) sum, we need to generate the carry from the 1's bit sum.

The carry plus the 2's bit value for each row comes to four source operands, whereas the blitter can only handle three at once. At this point, it seems we're running out of blits. Instead of going step-by-step through the rest of the algorithm, let me direct your attention to a summary in Table One.

Two remaining tricks enable the nine blit solution. The first trick (my contribution) is to smuggle the 1's bit carry in R5 into the 2's bit sum of the second row (blit 6). The ambiguity this move introduces is eliminated by referring to R5 again when performing blit 8. The second trick is to compute 2's-bit-sum-and-no-further-carry all at the same time (again, blit 8).

Use "B1," "B2," and "B3" to represent the bits in the total sum of a cell's neighbors (B1 = LSB, etc.), and "CC" for the current cell state. Since a cell is alive in the next generation ("NG") if, and only if, it has just three neighbors, or just two neighbors when currently alive, you get:

NG := (B1 and B2 and not-B3) or
(CC and not-B1 and B2 and not-B3);

By combining B2 and not-B3 (= 2's-bit-sum-and-no-further-carry) into a single term, J2 (for "just two"), and expanding, you get:

NG := (CC and B1 and J2) or (not-CC and B1 and J2)
or (CC and not-B1 and J2);

This formula conveniently requires only three source operands, and is the calculation performed by blit 9.

Listing One LIFE.ASM

```
*****
*
*      LIFE.ASM
*
* This routine is called by QBSBlit() in LIFER.C. It performs
* the number of blits passed to it in Lifebltnode.repeat by
* returning a non-zero value in D0, causing QBSBlit() to call it
* again.
*
* Requires linkage with the program in LIFER.C.
*
* Gerald Hull, CREATIVE FOCUS
*
*****

NOLIST
INCLUDE 'exec/types.i'
INCLUDE 'hardware/custom.i'
INCLUDE 'hardware/blit.i'
INCLUDE 'hardware/dmabits.i'
LIST

XDEF      _doblit
```

Public Domain Source Illustrating Blitter Use

Scott Everndon, LIFE: People Link public domain library. He uses the built-in AmigaDOS function, BltBitMap(), for calculating a Life generation. He is therefore limited to a single source bitplane in blitting. Nonetheless, this version contains very well structured C code and a nice user interface.

Alonzo Garipey, BLITTER HELP: 7/16/86 People Link notice. This is a useful discussion of how to use shifts with the blitter. Extremely illuminating fragments of his FASTLIFE program are included, showing, among other things, how to bang the registers.

Tomas Rokicki, BLITLAB: Fred Fish #84, Amicus #19, People Link library. A nice demo allowing the user to experiment with all sorts of blitter functions. The code also shows how to construct an elaborate gadget-laden interface.

Tomas Rokicki, LIFE: Fred Fish #31. This code presents an extremely fast version of Life in a program reduced to the essential elements. It contains the first publication, to my knowledge, of a ten blit solution.

```
STRUCTURE Lifebltnode,0
STRUCT      b_node,bn_SIZEOF
LONG        cons
LONG        apt
LONG        bpt
LONG        cpt
LONG        dpt
WORD        amod
WORD        bmod
WORD        cmod
WORD        dmod
WORD        afwm
WORD        alwm
WORD        repeat
WORD        size
WORD        dummy
LABEL       Lbn_SIZEOF
* C struct use longword boundaries
```

```
* In virtue of QBSBlit():
* a1 points to the first Lifebltnode
* a0 points to CUSTOM - $dfff000

_doblit
    moveq    #0,d1
    tst      repeat(a1)
    beq.s    2$,DZIP
    tst      DZIP
    bne.s    1$,repeat(a1),DZIP
1$    sub     #1,DZIP
2$    add     INDEX,d1
    move.l   cons(a1,d1),bltcon0(a0)
    move.l   apt(a1,d1),bltapt(a0)
    move.l   bpt(a1,d1),bltbpt(a0)
    move.l   cpt(a1,d1),bltcpt(a0)
    move.l   dpt(a1,d1),bltdpt(a0)
    move     amod(a1,d1),bltamod(a0)
    move     bmod(a1,d1),bltbmod(a0)
    move     cmod(a1,d1),bltcmmod(a0)
    move     dmod(a1,d1),bltdmod(a0)
    move     afwm(a1,d1),bltafwm(a0)
    move     alwm(a1,d1),bltalwm(a0)
3$    move     size(a1,d1),bltsize(a0)
    add      #Lbn_SIZEOF,INDEX
    moveq    #0,d0
    add      DZIP,d0
    bne.s    4$,INDEX
4$    rts

* DZIP is needed because line 3$ afflicts register d0
* both variables are 0 when blit sequence is complete

DZIP    dc.w    0
INDEX   dc.w    0

end
```


Listing Two LIFER.C

```

/
*****
LIFER.C

An Amiga implementation of John Conway's Game of Life. Much
of this code derives from similar programs by Alonzo Gariepy
and Scott Evernden. Feel free to similarly adapt it to your
own needs.

Requires linkage with the assembler routine _doblit listed
in LIFE.ASM.

Gerald Hull, CREATIVE FOCUS
*****

#include <exec/types.h>
#include <exec/exec.h>
#include <graphics/gfxbase.h>
#include <intuition/intuition.h>
#include <hardware/custom.h>
#include <hardware/blit.h>

struct Custom *chips = (struct Custom *) 0xDFF000;

typedef struct
{
    struct bltnode bnode;
    union
    {
        ULONG whole;
        struct
        {
            unsigned ashift : 4, usea : 1, useb : 1, usec : 1, used : 1;
            unsigned minterm : 8;
            unsigned bshift : 4, pad : 7, exfill : 1;
            unsigned infill : 1, fcarry : 1, descend : 1, line : 1;
        } part;
    } con;

    ULONG blt_apt, blt_bpt, blt_cpt, blt_dpt; /* bitplane ptrs */
    UWORD blt_amod, blt_bmod, blt_cmod, blt_dmod; /* offset in bytes */
    UWORD blt_afwm, blt_alwm; /* first and last word masks */
    UWORD repeat; /* number of blits; signal to asm */
    union
    {
        UWORD whole;
        struct
        {
            unsigned height : 10, width : 6;
        } part;
    } size;
} Lifebltnode;

#define LIFEBLITS 14
#define CRT LIFEBLITS
#define CLR LIFEBLITS+1

Lifebltnode lifeblt[LIFEBLITS+2];

#define makesize(a,b) (((a&0x03ff)<<6)+(b&0x003f))
#define blt_con con.whole
#define blt_usea con.part.usea
#define blt_useb con.part.useb
#define blt_usec con.part.usec
#define blt_used con.part.used
#define blt_minterm con.part.minterm
#define blt_ashift con.part.ashift
#define blt_bshift con.part.bshift
#define blt_size size.whole
#define blit_size bnode.blitsize.hole
#define blt_width size.part.width
#define blt_height size.part.height

/* The following are the different minterms for the life
generation */

#define ONEor3 0x96
#define TWOor3 0xE8
#define JUST_1 0x16
#define A_LIFE 0xC8
#define XOR 0x3C
#define TRIK_1 0x6A
#define TRIK_2 0x8C
#define ZIP 0x00

#define STARTNUM 1

void finish();
void init();
void size();
void menus();
void allocate_rasters();
void deallocate_rasters();
void life();
void generation();
void CritBlit();
void DoAblit();
void InitBlits();
void clrscreen();

extern ULONG doblit(); /* assembler routine called by QBSBlit() */

/* Screen/Window sizes */
#define DEPTH 1
#define SW 320 /* screen width */
#define SH 200 /* screen height */
#define WW SW /* window width */
#define WH SH /* window height */
#define BIGW WW+32 /* register bitplane width */
#define BIGH WH+10 /* register bitplane height */

#define WORDWIDTH (WW>>4)
#define BYTEWIDTH (WW>>3)

#define REGOFFSET 2+(BYTEWIDTH+4)*CellY

#define LIFEWIDTH WORDWIDTH
#define LIFEHEIGHT WH

/* Templates for the various cells */

struct
{
    WORD xsize, ysize;
    ULONG data[6];
    } critter[] =
    {
        { 1, 1, { 0x80000000 } },
        { 2, 2, { 0x80000000, 0x00000000 } },
        { 4, 4, { 0xE0000000, 0xE0000000, 0xE0000000, 0x00000000 } },
        { 5, 5, { 0x60000000, 0xF0000000, 0xD0000000, 0x60000000, 0x00000000 } }
    };

/* Needed structures */

struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
struct Window *w; /* window structure returned by exec */
struct Screen *s; /* screen structure returned by exec */

#define SCREEN 6
struct BitMap bmaps[SCREEN];
struct BitMap CellMap, ABitMap;
struct BitMap *bm[SCREEN+1];

struct NewScreen ns =
{
    0,
    0,
    SW,
    SH,
    DEPTH,
    0,
    1,
    0,
    CUSTOMSCREEN,
    NULL,
    NULL,
    NULL,
    NULL
};

struct NewWindow nw =
{
    0,
    0,
    WW,
    WH,
    0,
    1,
    MOUSEBUTTONS
    | MENUVERIFY
    | MENUPICK,
    BACKDROP
    | ACTIVATE
    | BORDERLESS,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL,
    NULL
}

```

(continued)


```

0,
0,
0,
0,
CUSTOMSCREEN
};

/* Menu stuff */

#define MENUS 2
#define M1_ITEMS 4
#define M2_ITEMS 4

struct Menu Mmenu[MENUS];
struct MenuItem M1Items[M1_ITEMS];
struct MenuItem M2Items[M2_ITEMS];
struct IntuiText MTexts[M1_ITEMS];
struct Image MImages[M2_ITEMS];

WORD im_wid[M2_ITEMS] = {5, 7, 13, 16};
USHORT image[M2_ITEMS][16] =
{
{ 0x0000,
  0x2000,
  0x1000,
  0x7000
},
{ 0x0000,
  0x1000, 0x0000,
  0x0400, 0x0000,
  0x5400
},
{ 0x0000,
  0x0700, 0x0700, 0x0700, 0x0000,
  0x0070, 0x0070, 0x0070, 0x0000,
  0x7770, 0x7770, 0x7770
},
{ 0x0000,
  0x0180, 0x03C0, 0x0340, 0x0180, 0x0000,
  0x000C, 0x001E, 0x001A, 0x000C, 0x0000,
  0x318C, 0x7BDE, 0x6B5A, 0x318C
}
};

LONG CellX, CellY;
ULONG screen, thisgen;
ULONG reg1, reg2, reg3, reg4, reg5;

/******

void finish()

/* Release resources
*/

{
if (w)
CloseWindow(w);
if (s)
CloseScreen(s);
if (GfxBase)
CloseLibrary(GfxBase);
if (IntuitionBase)
CloseLibrary(IntuitionBase);
}

void init()

/* Open libs, screen & window, fix colors, etc.
*/

{
GfxBase = (struct GfxBase *)
OpenLibrary("graphics.library", 0);
if (!GfxBase)
{
puts("no graphics!!!");
finish();
exit(20);
}

IntuitionBase = (struct IntuitionBase *)
OpenLibrary("intuition.library", 0);
if (!IntuitionBase)
{
puts("no intuition!!!");
finish();
exit(21);
}

s = (struct Screen *) OpenScreen(&ns);
if (!s)
{
puts("no screen!!!");
finish();
}
}

```

```

exit(22);
}

ShowTitle(s, FALSE);
SetRGB4(&s->ViewPort, 0, 12, 12, 12);
SetRGB4(&s->ViewPort, 1, 0, 0, 0);
bm[SCREEN] = &AbitMap;
InitBitMap(bm[SCREEN], DEPTH, WW, WH);
nw.Screen = s;

w = (struct Window *) OpenWindow(&nw);
if (!w)
{
puts("no window");
finish();
exit(23);
}

bm[SCREEN] = w->RPort->BitMap;
screen = (ULONG)bm[SCREEN]->Planes[0];
}

void sizes(which)
WORD which;

/* Calculate assorted parameters from cell size
*/

{
CellX = critter[which].xsize;
CellY = critter[which].ysize;

InitBitMap(&CellMap, 1, CellX, CellY);
CellMap.Planes[0] = (PLANEPTR)critter[which].data;

InitBlits(which);
}

void menus()

/* Set 'em up
*/

{
WORD i, t;

Mmenu[0].NextMenu = &Mmenu[1];
Mmenu[0].LeftEdge = 4;
Mmenu[0].TopEdge = 0;
Mmenu[0].Width = 58;
Mmenu[0].Height = 10;
Mmenu[0].Flags = MENUENABLED;
Mmenu[0].MenuName = "Actions";
Mmenu[0].FirstItem = &M1Items[0];

for (i = 0; i < M1_ITEMS; i++)
{
MTexts[i].FrontPen = 0;
MTexts[i].BackPen = 1;
MTexts[i].DrawMode = JAM2;
MTexts[i].LeftEdge = 0;
MTexts[i].TopEdge = 1;
MTexts[i].ITextFont = NULL;
MTexts[i].NextText = NULL;

M1Items[i].NextItem =
i == M1_ITEMS-1 ? NULL : &M1Items[i+1];
M1Items[i].LeftEdge = 0;
M1Items[i].Width = 115;
M1Items[i].Height = 10;
M1Items[i].Flags = ITEMTEXT | ITEMENABLED | HIGHCOMP;
if (i && i != 3)
M1Items[i].Flags |= CHECKIT;
M1Items[i].ItemFill = (APTR) &MTexts[i];
M1Items[i].SelectFill = NULL;
M1Items[i].Command = 0;
M1Items[i].SubItem = NULL;
M1Items[i].NextSelect = 0;
}

MTexts[0].IText = " Clear Screen";
M1Items[0].TopEdge = 0;

MTexts[1].IText = " Go";
M1Items[1].TopEdge = 12;
M1Items[1].MutualExclude = 4;

MTexts[2].IText = " Stop";
M1Items[2].TopEdge = 24;
M1Items[2].Flags |= CHECKED;
M1Items[2].MutualExclude = 2;

MTexts[3].IText = " Quit";
M1Items[3].TopEdge = 36;

Mmenu[1].NextMenu = NULL;
Mmenu[1].LeftEdge = 70;
}

```



```

Mmenu[1].TopEdge = 0;
Mmenu[1].Width = 42;
Mmenu[1].Height = 10;
Mmenu[1].Flags = MENUEENABLED;
Mmenu[1].MenuName = "Cells";
Mmenu[1].FirstItem = &M2Items[0];

for (i = 0; i < M2_ITEMS; i++)
{
    t = im_wid[i];
    MImages[i].LeftEdge = 5 + LOWCHECKWIDTH;
    MImages[i].TopEdge = (4 + im_wid[M2_ITEMS-1] - t) / 2;
    MImages[i].Width = t;
    MImages[i].Height = t;
    MImages[i].Depth = DEPTH;
    MImages[i].ImageData = (USHORT *) &image[i][0];
    MImages[i].PlanePick = 1;
    MImages[i].PlaneOnOff = 0;
    MImages[i].NextImage = NULL;

    t = 4 + im_wid[M2_ITEMS-1];
    M2Items[i].NextItem
        = i == M2_ITEMS-1 ? NULL : &M2Items[i+1];
    M2Items[i].LeftEdge = 0;
    M2Items[i].TopEdge = i * t;
    M2Items[i].Width = 5 + LOWCHECKWIDTH + t;
    M2Items[i].Height = t;
    M2Items[i].Flags = ITEMENABLED | HIGHBOX | CHECKIT;
    if (i == STARTNUM)
        M2Items[i].Flags |= CHECKED;
    M2Items[i].MutualExclude = ~(1L << i);
    M2Items[i].ItemFill = (APTR) &MImages[i];
    M2Items[i].SelectFill = NULL;
    M2Items[i].Command = 0;
    M2Items[i].SubItem = NULL;
    M2Items[i].NextSelect = 0;
}

void allocate_rasters()
/* Get the rasters
*/
{
    int i;

    for (i = 0; i < SCREEN; i++)
    {
        bm[i] = &bmaps[i];
        InitBitMap(bm[i], 1, WW+32, WH+10);
        bmaps[i].Planes[0]
            = (PLANEPTR)AllocRaster(WW+32, WH+10);
        if (!bmaps[i].Planes[0])
        {
            puts("no memory!");
            deallocate_rasters();
            finish();
            exit(25);
        }
    }

    thisgen = (ULONG)bm[0]->Planes[0];
    reg1 = (ULONG)bm[1]->Planes[0];
    reg2 = (ULONG)bm[2]->Planes[0];
    reg3 = (ULONG)bm[3]->Planes[0];
    reg4 = (ULONG)bm[4]->Planes[0];
    reg5 = (ULONG)bm[5]->Planes[0];
}

void deallocate_rasters()
/* Free the rasters
*/
{
    int i;

    for (i = 0; i < SCREEN; i++)
        if (bmaps[i].Planes[0])
            FreeRaster(bmaps[i].Planes[0], WW+32, WH+10);
}

/*****/

void _main() /* Toebe's trick to eliminate startup (c.o) */

/* This is it, what it is.
*/
{
    init();
    menus();
    SetMenuStrip(w, &Mmenu[0]);
    allocate_rasters();
    sizes(STARTNUM);
    life();
    deallocate_rasters();
}

```

```

ClearMenuStrip(w);
finish();
}

/*****/

void life()
/* Parse input, behave accordingly
*/
{
    struct MenuItem *m_item;
    struct IntuiMessage *message;
    ULONG class;
    USHORT code;
    SHORT item, menu, mx = 1, my = 1, wasmx = 0, wasmy = 0;
    WORD quitting, running, mousing, newdown, menuing;

    quitting = running = mousing = menuing = FALSE;

    while (!quitting)
    {
        if (!(running || mousing))
            Wait(1L << w->UserPort->mp_SigBit);

        while (message = (struct IntuiMessage *)
            GetMsg(w->UserPort))
        {
            class = message->Class;
            code = message->Code;

            ReplyMsg(message);

            switch (class)
            {
                case MENUPICK:
                    menuing = FALSE;
                    while (code != MENUNULL)
                    {
                        menu = MENUNUM(code);
                        item = ITEMNUM(code);

                        if (menu == 0 && item == 3)
                        {
                            quitting = TRUE;
                            break;
                        }
                        else if (menu == 1 || item == 0)
                        {
                            DoABlit(&lifeblt[CLR]);
                            MItems[1].Flags &= ~CHECKED;
                            MItems[2].Flags |= CHECKED;
                            running = FALSE;
                            if (menu == 1)
                                sizes(item);
                        }
                        else if (item < 3)
                            running = item == 1;

                        m_item = (struct MenuItem *)
                            ItemAddress(&Mmenu[0], code);
                        code = m_item->NextSelect;
                    }
                    break;

                case MENUVERIFY:
                    menuing = TRUE;
                    break;

                case MOUSEBUTTONS:
                    mousing = code == SELECTDOWN;
                    newdown = mousing;
                    break;
            }
        }

        if (menuing || quitting)
            continue;

        if (mousing)
        {
            mx = CellX * ((w->MouseX) / CellX);
            my = CellY * ((w->MouseY) / CellY);
            if (mx+CellX <= WW && my+CellY <= WH)
            {
                if (newdown || wasmx != mx || wasmy != my)
                {
                    CritBlit(mx, my);
                    newdown = FALSE;
                    wasmx = mx;
                    wasmy = my;
                }
            }
            else if (running)

```

(continued)


```

        DoABlit(&lifeblt[0]);
    }
}

void DoABlit(lifeblt)
    Lifebltnode *lifeblt;

/* This is where the blitter gets invoked */
{
    OwnBlitter();
    WaitBlit();
    QBSBlit(lifeblt);
    WaitBlit();
    DisownBlitter();
}

void CritBlit(mx, my)
    SHORT mx, my;

/* This XORs the different sized cells ("critters") onto the screen */
{
    lifeblt[CRT].blt_ashift = mx & 16;
    lifeblt[CRT].blt_bpt =
        screen + (my * BYTEWIDTH) + (mx / 16) * 2;
    lifeblt[CRT].blt_dpt =
        screen + (my * BYTEWIDTH) + (mx / 16) * 2;
    DoABlit(&lifeblt[CRT]);
}

void InitBlits(which)
    WORD which;

/* Set up the parameters that are the same for almost all the
blits. Since the Lifebltnode externals are initialized to 0 we
needn't worry about null parameters. */
{
    int i, bn;

    for (i = 0; i < LIFEBLITS+2; ++i)
    {
        lifeblt[i].bnode.function = doblit;
        lifeblt[i].bnode.blitsize
            = makesize(LIFEHEIGHT, WORDWIDTH);
        lifeblt[i].blt_usea = 1;
        lifeblt[i].blt_useb = 1;
        lifeblt[i].blt_usec = 1;
        lifeblt[i].blt_used = 1;
        lifeblt[i].blt_afwm = 0xFFFF;
        lifeblt[i].blt_alwm = 0xFFFF;
        lifeblt[i].blt_width = WORDWIDTH;
        lifeblt[i].blt_height = LIFEHEIGHT;
        lifeblt[i].repeat = LIFEBLITS;
    }

    /* First we do five blits to enable wrap-around processing
ALONZO) */
    bn = 0; /* 1 */

    lifeblt[bn].blt_useb = 0;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt = screen;
    lifeblt[bn].blt_dpt = thisgen + REGOFFSET;
    lifeblt[bn].blt_dmod = 4;
    lifeblt[bn].blt_minterm = A_TO_D;

    ++bn; /* 2 */

    lifeblt[bn].blt_useb = 0;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt
        = thisgen + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_dpt
        = thisgen + (BYTEWIDTH + 4) * (LIFEHEIGHT + CellY);
    lifeblt[bn].blt_minterm = A_TO_D;
    lifeblt[bn].blt_width = WORDWIDTH + 2;
    lifeblt[bn].blt_height = CellY;
    lifeblt[bn].bnode.blitsize = makesize(CellY, WORDWIDTH+2);

    ++bn; /* 3 */

    lifeblt[bn].blt_useb = 0;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt
        = thisgen + (BYTEWIDTH + 4) * LIFEHEIGHT;
    lifeblt[bn].blt_dpt = thisgen;
    lifeblt[bn].blt_minterm = A_TO_D;
    lifeblt[bn].blt_width = WORDWIDTH + 2;
    lifeblt[bn].blt_height = CellY;
    lifeblt[bn].bnode.blitsize = makesize(CellY, WORDWIDTH+2);

```

```

    ++bn; /* 4 */

    lifeblt[bn].blt_useb = 0;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt = thisgen + 2;
    lifeblt[bn].blt_amod = BYTEWIDTH + 2;
    lifeblt[bn].blt_dpt = thisgen + BYTEWIDTH + 2;
    lifeblt[bn].blt_dmod = BYTEWIDTH + 2;
    lifeblt[bn].blt_minterm = A_TO_D;
    lifeblt[bn].blt_width = 1;
    lifeblt[bn].blt_height = LIFEHEIGHT + 2 * CellY;
    lifeblt[bn].bnode.blitsize
        = makesize(LIFEHEIGHT+2*CellY, 1);

    ++bn; /* 5 */

    lifeblt[bn].blt_useb = 0;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt = thisgen + BYTEWIDTH;
    lifeblt[bn].blt_amod = BYTEWIDTH + 2;
    lifeblt[bn].blt_dpt = thisgen;
    lifeblt[bn].blt_dmod = BYTEWIDTH + 2;
    lifeblt[bn].blt_minterm = A_TO_D;
    lifeblt[bn].bnode.blitsize
        = makesize(LIFEHEIGHT+2*CellY, 1);
    lifeblt[bn].blt_width = 1;
    lifeblt[bn].blt_height = LIFEHEIGHT + 2 * CellY;

    /* Now nine economical blits to do a new generation */
    ++bn; /* 6 */

    lifeblt[bn].blt_ashift = CellX;
    lifeblt[bn].blt_bshift = 16 - CellX;
    lifeblt[bn].blt_apt = thisgen;
    lifeblt[bn].blt_amod = 2;
    lifeblt[bn].blt_bpt = thisgen + 2;
    lifeblt[bn].blt_bmod = 2;
    lifeblt[bn].blt_cpt = thisgen;
    lifeblt[bn].blt_cmmod = 2;
    lifeblt[bn].blt_dpt = reg1;
    lifeblt[bn].blt_dmod = 2;
    lifeblt[bn].blt_minterm = ONEor3;
    lifeblt[bn].blt_width = LIFEWIDTH + 1;
    lifeblt[bn].blt_height = LIFEHEIGHT + 2 * CellY;
    lifeblt[bn].bnode.blitsize
        = makesize(LIFEHEIGHT+2*CellY, LIFEWIDTH+1);

    ++bn; /* 7 */

    lifeblt[bn].blt_ashift = CellX;
    lifeblt[bn].blt_bshift = 16 - CellX;
    lifeblt[bn].blt_usec = 0;
    lifeblt[bn].blt_apt = thisgen + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_amod = 2;
    lifeblt[bn].blt_bpt = thisgen + 2 + (BYTEWIDTH + 4) *
CellY;
    lifeblt[bn].blt_bmod = 2;
    lifeblt[bn].blt_dpt = reg2 + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_dmod = 2;
    lifeblt[bn].blt_minterm = XOR;
    lifeblt[bn].blt_width = LIFEWIDTH + 1;
    lifeblt[bn].bnode.blitsize = makesize(LIFEHEIGHT, LIFEWIDTH+1);

    ++bn; /* 8 */

    lifeblt[bn].blt_apt = reg1 + 2;
    lifeblt[bn].blt_amod = 4;
    lifeblt[bn].blt_bpt = reg2 + 2 + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_bmod = 4;
    lifeblt[bn].blt_cpt = reg1 + 2 + (BYTEWIDTH + 4) * 2 *
CellY;
    lifeblt[bn].blt_cmmod = 4;
    lifeblt[bn].blt_dpt = reg4 + 2 + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_dmod = 4;
    lifeblt[bn].blt_minterm = ONEor3;

    ++bn; /* 9 */

    lifeblt[bn].blt_apt = reg1 + 2;
    lifeblt[bn].blt_amod = 4;
    lifeblt[bn].blt_bpt = reg2 + 2 + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_bmod = 4;
    lifeblt[bn].blt_cpt = reg1 + 2 + (BYTEWIDTH + 4) * 2 *
CellY;
    lifeblt[bn].blt_cmmod = 4;
    lifeblt[bn].blt_dpt = reg5 + 2 + (BYTEWIDTH + 4) * CellY;
    lifeblt[bn].blt_dmod = 4;
    lifeblt[bn].blt_minterm = TWOor3;

    ++bn; /* 10 */

```


TxED Plus... All the speed and simplicity of the original TxED, Plus...

Modular software is coming. With modular software, different programs talk to each other using a common macro language. Apple Computer, Inc. has started a software division focusing on modular software, and Microsoft Inc's Bill Gates has been talking about it. Modular software lets "multitasking" mean more than just running two programs at the same time; you can run programs *together*, doing more than each program can do alone and letting you pick exactly the pieces you want to use.

On the Amiga, modular applications are more than just next year's dream. They're available now, with two of the cornerstones ready to go. TxED Plus, the text editor, and AREXX, the macro processor.

Even without the AREXX connection, TxED Plus is the text editor of choice for the Amiga. With AREXX, TxED Plus becomes more than just a text editor. The configurable menus allow you to create customized applications such as order entry systems, and that's barely scratching the surface of the possibilities.

We don't even know what the limits are yet.

Developers: find out about AREXX! The window is wide open.

Users: demand AREXX capability in your software. You can get it now on your Amiga, or wait till next year, on a McClone.

- Fully configurable menus and keyboard
- Powerful command line language, uses the AREXX macro processor.
- Includes functional AREXX demo
 - Many other new features.
 - Uses ARP V1.1

TxED Plus \$79.95



Microsmiths, Inc

PO Box 561, Cambridge, MA 02140

(617) 354-1224

BIX: cheath CIS: 74216,2117

Mass Residents add 5%. Visa & MC accepted.

Amiga is a trademark of Commodore Business Machines, Inc.

The term 'McClone' is a fictitious conglomeration referring to nothing in particular.

```
lifeblt[bn].blt_ashift = CellX;
lifeblt[bn].blt_bshift = 16 - CellX;
lifeblt[bn].blt_apt = thisgen;
lifeblt[bn].blt_amod = 2;
lifeblt[bn].blt_bpt = thisgen + 2;
lifeblt[bn].blt_bmod = 2;
lifeblt[bn].blt_cpt = thisgen;
lifeblt[bn].blt_cmod = 2;
lifeblt[bn].blt_dpt = reg1;
lifeblt[bn].blt_dmod = 2;
lifeblt[bn].blt_minterm = TWOor3;
lifeblt[bn].blt_width = LIFEWIDTH + 1;
lifeblt[bn].blt_height = LIFEHEIGHT + 2 * CellY;
lifeblt[bn].bnode.blitsize = makesize(LIFEHEIGHT+2*CellY,
LIFEWIDTH+1);

++bn; /* 11 */

lifeblt[bn].blt_ashift = CellX;
lifeblt[bn].blt_bshift = 16 - CellX;
lifeblt[bn].blt_apt = thisgen + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_amod = 2;
lifeblt[bn].blt_bpt = thisgen + 2 + (BYTEWIDTH + 4) *
CellY;
lifeblt[bn].blt_bmod = 2;
lifeblt[bn].blt_cpt = reg5 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_cmod = 2;
lifeblt[bn].blt_dpt = reg2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_dmod = 2;
lifeblt[bn].blt_minterm = TRIK_1;
lifeblt[bn].blt_width = LIFEWIDTH + 1;
lifeblt[bn].bnode.blitsize = makesize(LIFEHEIGHT, LIFEWIDTH+1);

++bn; /* 12 */

lifeblt[bn].blt_apt = reg1 + 2;
lifeblt[bn].blt_amod = 4;
lifeblt[bn].blt_bpt = reg2 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_bmod = 4;
lifeblt[bn].blt_cpt = reg1 + 2 + (BYTEWIDTH + 4) * 2 *
CellY;
lifeblt[bn].blt_cmod = 4;
lifeblt[bn].blt_dpt = reg3 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_dmod = 4;
lifeblt[bn].blt_minterm = JUST_1;
```

```
++bn; /* 13 */

lifeblt[bn].blt_apt = reg5 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_amod = 4;
lifeblt[bn].blt_bpt = reg3 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_bmod = 4;
lifeblt[bn].blt_cpt = reg2 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_cmod = 4;
lifeblt[bn].blt_dpt = reg1 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_dmod = 4;
lifeblt[bn].blt_minterm = TRIK_2;

++bn; /* 14 */

lifeblt[bn].blt_apt = thisgen + 2 + (BYTEWIDTH + 4) *
CellY;
lifeblt[bn].blt_amod = 4;
lifeblt[bn].blt_bpt = reg1 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_bmod = 4;
lifeblt[bn].blt_cpt = reg4 + 2 + (BYTEWIDTH + 4) * CellY;
lifeblt[bn].blt_cmod = 4;
lifeblt[bn].blt_dpt = screen;
lifeblt[bn].blt_minterm = A_LIFE;

/* Finally, two special blits for the user interface */

/* CRT */

lifeblt[CRT].blt_width = 2;
lifeblt[CRT].blt_height = 6;
lifeblt[CRT].blt_usec = 0;
lifeblt[CRT].blt_apt = (ULONG)(&critter[which].data);
lifeblt[CRT].blt_bmod = BYTEWIDTH - 4;
lifeblt[CRT].blt_dmod = BYTEWIDTH - 4;
lifeblt[CRT].blt_minterm = XOR;
lifeblt[CRT].bnode.blitsize = makesize(6,2);
lifeblt[CRT].repeat = 0;

/* CLR */

lifeblt[CLR].blt_usec = 0;
lifeblt[CLR].blt_apt = 0;
lifeblt[CLR].blt_minterm = ZIP;
lifeblt[CLR].repeat = 0;
lifeblt[CLR].blt_apt = screen;
lifeblt[CLR].blt_dpt = screen;
}
```

•AC•

by John Steiner

Bug Bytes

The Bugs & Upgrades Column

I have many reports this month from readers who have found bugs in various software packages. I am especially interested in finding workarounds to bug problems.

At the last Fargo-Moorhead Amiga Users Group meeting, Carl Eidbo, a recent purchaser of an Amiga 500, brought in an AmigaBASIC program he was working on, and asked members why it constantly caused his Amiga to Guru. Nothing was visibly wrong with the code, so we narrowed it down to just three lines. The program listing below ensures a visit from the Guru:

```
NS="Welcome Mr. Guru"  
SAY TRANSLATES(NS) BEEP  
SAY TRANSLATES(NS)
```

We compiled the program with the AC-BASIC compiler, and it worked flawlessly. We also found that any slight delay causing the program to pause before executing the last SAY command avoids the Guru.

Coincidentally, I received a letter from Jerry Vaughn of San Diego, CA, reporting the same bug. He noted that the problem does not appear under AmigaBASIC 1.1. His code is similar to Mr. Eidbo's, except the BEEP command is replaced by the SOUND command.

Mr. Vaughn sent along copies of correspondence he has sent to Microsoft and Commodore about the problem. Microsoft responded with a letter stating that since AmigaBASIC is licensed to Commodore, Commodore

provides software support. The letter to Commodore, dated March 25, 1987, is still unanswered.

Ronald Mushin, of Downey, CA, reported that Mirror version 1.5 has a bug that prevents it from running on some Amiga A2000 computers. Programs from other companies may also have difficulty because this problem can be traced to a change in the keyboard design on the new machines. Mirror self-boots properly and brings up the introductory screen, according to Mr. Mushin, but when you try to select either F1 or F2, the keyboard locks up and will no longer communicate with the computer.

Frank Lewis of Holly Springs, NC, wrote with some workarounds for ABSOFT's AC-BASIC Compiler. Bryan Catley's review of AC-BASIC (AC V2.10) mentioned some of these bugs.

If a program has been compiled without the "U" option or DIM STATIC, and is then recompiled with either of these options, the stack size information in the icon file ON DISK gets updated correctly. This is not a bug in AC-BASIC, but a quirk of the Workbench system. The icon already displayed on the screen is a memory-resident copy of the original, however, and has not been changed. The solution involves forcing Workbench to re-read the icon from disk after re-compiling. If the .run icon is in a drawer, close and re-open that drawer's window; if the icon appears in the disk window, close all open drawer windows for that disk, and then close and re-open the disk window.

The CVL function can produce all kinds of bizarre and undesirable results (including visits to Mr. Guru). As reported in an earlier column, a patch is available from Absoft.

When doing library calls, the current version of the compiler does not properly support the requirements of expanded memory systems. Programs that open icon.library or graphics.library (and others, perhaps?) will run successfully, but some nasty things may happen later, including lost icons, Guru meditations, and system lock-ups. FixHunk, a public domain utility, patches all memory allocation requests to CHIP memory only and solves the problem.

Mr. Lewis also reported that compiled programs using arrays (such as GET/PUT and BOB arrays) to hold graphics data are likely to display the graphic as blank or garbage on expanded-memory systems. Patching the .run file solves this problem if the arrays aren't STATIC (either through DIM STATIC or compiling with the "U" option). According to Mr. Lewis, since the STATIC arrays are allocated in the stack area, and there isn't any way to force AmigaDOS to allocate a stack in CHIP memory, this is the only possible workaround.

Charles Followell of Franklin, IN, reported a problem with the Zing! documentation. Mr. Followell runs an A1000 with 1.5 meg of RAM and 1.2 KS/WorkBench.

While configuring ZING!, he tried to set the default gadgets at the bottom of the "File Information Window." The manual states that you need only use the abbreviation DSK to assign the Diskcopy window. Every time he tried, the program responded with "invalid abbreviation." After ordering an updated manual, he found that even the new manual listed the DSK instruction.

Using Newzap, Mr. Followell scanned the disk and found a list of all three letter abbreviations. DSK is not listed, but DCP is listed. The workaround for this one is to simply use DCP, instead of DSK. Mr. Followell also comments that ZING! is a great product.

My son, Joshua, is a fan of Barbarian, and he plays it regularly. He was particularly proud to report a bug he found in the program. He quite innocently asked me why Barbarian goes off to Guru-Land every time our phone rings. After running a couple of tests, we confirmed the bug. My Amiga 1680 modem is connected to the computer serial port and the telephone line. Evidently the ring detect signal from the modem causes the problem. The problem occurred on both the A500 and A2000.

Speaking of the Amiga 2000, my keyboard has a problem sensing the first key press after reboot. Upon cold-start or re-boot, the first key press is ignored. At first I thought I was not hitting a key properly, but I have found the problem occurred one hundred percent of the time. The newest keyboard for the Amiga 2000 seems to be the culprit. The latest shipment of A2000 computers was packaged differently from earlier shipments, and the entire shipment exhibits the problem. The new keyboards look identical to the old ones, but they are held together with phillips screws, rather than plastic holding tabs.

WordPerfect is shipping an upgrade dated November 25, 1987. You can

find your version's date by pressing the Help key; the version date is in the menu bar at the right. The update, available directly from WordPerfect, fixes several bugs. A major overhaul of the Okimate 20 printer driver is included. The Text In/Out menu has added Save and Load IBM Word Perfect file options, and Save and Load IBM Text file options.

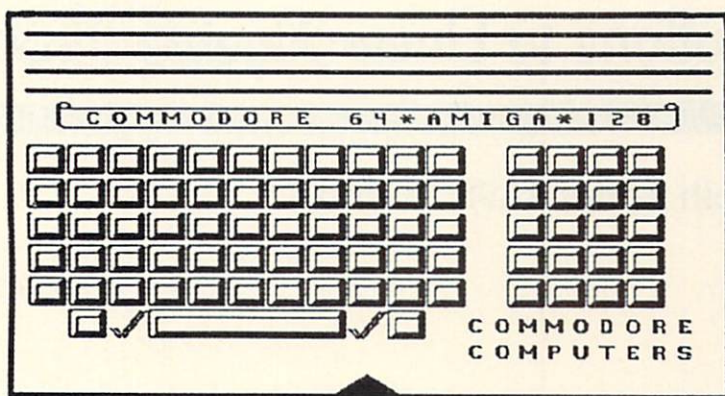
Support for Amiga Intuition style keystrokes has also been added. These keys work with many packages that support Intuition. Here is a list of alternate WP commands: Pressing the left Amiga key and I selects the next word to the right of the cursor; O selects the next sentence to the right of the cursor; P selects the next paragraph to the right of the cursor; J selects the next word to the left of the cursor; K selects the next sentence to the left of the cursor; L selects the next paragraph to the left of the cursor; N brings Workbench to the front; M

sends Workbench to the back. Pressing the right Amiga key and X cuts a block; C copies a block; P pastes a block; I changes the attribute to italics or returns it to plain text; B changes the attribute to bold or returns it to plain text; U changes the attribute to underline or back to plain text; Q cancels (same as F1); and S saves a file (same as F10).

To receive an update, visit your WordPerfect Dealer, who can make updated copies for you (if you bring in your master disks) or contact Word Perfect and report the bug. Updates to repair bugs are free.

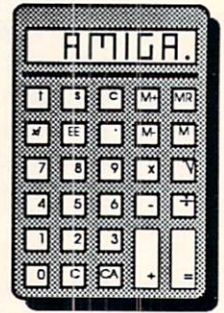
WordPerfect Corporation 288 West Center Street Orem, Utah 84057
Customer Support (800) 321-5906

Send all correspondence to:
John Steiner
c/o Amazing Computing
P.O. Box 869
Fall River, MA 02720



Solutions to Linear Algebra through Matrix Computations

in both AmigaBASIC™ and C



by Robert Ellis

This article details the basic operations and computer routines necessary for solving linear equations using matrix algebra. The routines that must be developed for matrix algebra will perform the following operations on a given matrix or set of matrices:

- addition (mat_add())
- subtraction (mat_sub())
- multiplication (mat_mlt())
- determinant of a mtri (mat_det())
- Inverse of a matrix (mat_inv())
- copy of a matrix (mat_cpy())
- multiplication of a matrix by a scalar (mat_scl())

Definition of Terms

ROW—A row is a set of matrix elements which occupy the elements running horizontally in the matrix. In the example matrix below, ROW 0 has its elements in capital letters.

$$\begin{bmatrix} A1,1 & A1,2 \\ a2,1 & a2,2 \end{bmatrix}$$

COLUMN—A column is a set of matrix elements which occupy the elements running vertically in the matrix. In the example matrix below, COLUMN 0 has its elements in capital letters.

$$\begin{bmatrix} A1,1 & a2,1 \\ A1,2 & a2,2 \end{bmatrix}$$

ORDER—The order of a matrix is defined as the ROW by COLUMN size of a matrix. This number can be calculated by counting the number of rows and columns, or by the position of the lower right hand element. The examples above are matrices of order (2x2). The following matrix has an order of (2 x 3).

$$\begin{bmatrix} a1,1 & a1,2 & a1,3 \\ a2,1 & a2,2 & a2,3 \end{bmatrix}$$

NOTE: The element values in these examples start at 1. The implementation of the computer routines have index values starting at a value of 0, however. This difference allows more efficient memory storage allocation.

Matrix Addition (mat_add())

Forming the sum of two matrices (A and B) to result in a third matrix (C) is a simple operation. This operation requires only that the two matrices are of the same order. (The number of rows in matrix A must equal the number of rows in matrix B, and the number of columns in matrix A equals the number of columns in matrix B.) The number of rows in each need not equal the number of columns, however.

The three primary routines are: the formation of an inverse matrix, determinant of a matrix, and multiplication of two matrices. The basic operations for solving linear equations can be performed with these three routines. The additional routines are provided as support routines.

Solving linear equations is important to electrical engineering. Linear equations can be used to describe a variety of problems dealing with electrical network analysis, transient analysis problems, and sensitivity and tolerance analysis. The engineer can become more accurate and productive using these matrix tools.

The algorithms and basic matrix operations are discussed in this article, and the implementations of these operations are included to help you use matrix algebra effectively. The implementations of the computer routines are written in C and placed in a single library file. The matrix operations are also implemented in AmigaBASIC. With all the C routines abstracted into one library, it is easier for the programmer to use the tools effectively. The BASIC version is included (because of the manner in which the subroutines share variables) in the file which contains the main program.

The matrix sum is produced by adding each individual element of matrix A to the equivalent element in matrix B. The result is placed in a third matrix (matrix C), as shown in the following example:

$$A = \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} \quad B = \begin{vmatrix} 2 & 1 \\ 1 & 3 \end{vmatrix}$$

order = (2 x 2) order = (2 x 2)

$$(A) + (B) = (C) = \begin{vmatrix} 3 & 4 \\ 6 & 10 \end{vmatrix}$$

order = (2 x 2)

The C routine to perform this operation is comprised of two loops. The outer loop cycles through the set of ROW index values. The inner loop cycles through the set of COLUMN index values. At each unique index pair, the element of matrix A is added to the element of matrix B. The result is placed in the C output matrix.

Matrix Subtraction (mat_sub())

Forming the difference of two matrices is identical to the matrix addition process, with one notable exception: subtraction is the operation at the element level. Example:

$$A = \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} \quad B = \begin{vmatrix} 2 & 1 \\ 1 & 3 \end{vmatrix}$$

order = (2 x 2) order = (2 x 2)

$$(A) - (B) = (C) = \begin{vmatrix} -1 & 2 \\ 4 & 4 \end{vmatrix}$$

order = (2 x 2)

Matrix Scalar (mat_scl())

Scalar multiplication of a matrix is a very simple operation. Each element of the matrix is independently multiplied by the external scalar value. The routine which performs this action (mat_scl()) is comprised of two loops. The outer loop indexes the ROW values and the inner loop indexes the COLUMN values. At each location in the matrix, a floating point multiplication of the element and the scalar takes place. The result is placed in an output matrix. Example:

$$A = \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} \quad \text{scalar} = 3$$

order = (2 x 2)

$$(A) * \text{scalar} = (B) = \begin{vmatrix} 3 & 9 \\ 15 & 21 \end{vmatrix}$$

order = (2 x 2)

Matrix Multiplication (mat_mlt())

Matrix multiplication is restricted by the size of the two input matrices. The restriction for operation $A * B$ is that the number of COLUMNS in A equals the number of ROWs in B; therefore the operation $A * B$ is NOT equal to operation $B * A$. Example:

$$A = \begin{vmatrix} 1 & 3 \\ 5 & 7 \end{vmatrix} \quad B = \begin{vmatrix} 2 & 1 \\ 1 & 3 \end{vmatrix}$$

order = (2 x 2) order = (2 x 2)

$$c1,1 = (a1,1 * b1,1) + (a1,2 * b2,1)$$

$$5 = (1 * 2) + (3 * 1)$$

$$c1,2 = (a1,1 * b1,2) + (a1,2 * b2,2)$$

$$10 = (1 * 1) + (3 * 3)$$

$$c2,1 = (a2,1 * b1,1) + (a2,2 * b2,1)$$

$$17 = (5 * 2) + (7 * 1)$$

$$c2,2 = (a2,1 * b1,2) + (a2,2 * b2,2)$$

$$26 = (5 * 1) + (7 * 3)$$

$$(A) * (B) = (C) = \begin{vmatrix} 5 & 10 \\ 17 & 26 \end{vmatrix}$$

order = (2 x 2)

Matrix Determinant (mat_det())

Finding the determinant of a matrix reduces the elements of a square matrix to a single scalar value. This operation was credited to G. W. Leibnitz in 1693 and later updated by G. Cramer in 1750. Judicious use of these determinants can help you solve simultaneous equations for independent variables.

Finding the determinant is limited by the fact that the matrix being operated on must be a square matrix. For example, the syntax of the determinant in an equation would for example read $\text{Det}(A) = 5$.

The matrix has a set of properties and restrictions limiting the types of matrices which can be used to form a determinant value. The first of the restrictions has already been stated—the matrix must be a square matrix. This means the number of rows must be equal to the number of columns. The other special properties are as follows:

- 1) If any row or column of the input matrix contains all zero values, the resulting determinant will have a value of zero.
- 2) If any two rows or columns of the input matrix are proportional to each other, the resulting determinant will have a zero value.

(continued)

3) If any full row or column of the input matrix is multiplied by a scalar value, the resulting determinant will be that many scalar value times larger than the original determinant value.

4) If the determinant of an input matrix has a set resulting value, and if two rows or columns of that input matrix are interchanged, the resulting determinant value will equal the negative of the original scalar value.

The computer routine in this article is based on "Gauss's elimination method." Another method, although not discussed here, involves evaluating the minors of a matrix and sum of the results. The total number of calculations using the Gauss method yields a result in $(N^3)/3$ operations.

The Gauss method is based on a technique that uses a "pivot" element as a reference point. This method aims at forming a resulting matrix which has all zero values to the lower left of the major diagonal. An example of this method:

$$A = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{vmatrix}$$

Once the matrix is in the form shown above, the value of the determinant is formed by multiplying all the elements along the major diagonal ($A = a_{11} * a_{22} * a_{33}$).

The following set of steps describes the actions taken by the computer routine to solve the determinant. The steps are written in an algorithm which can be implemented in any language.

- [STEP 1] Select a pivot element.
- [STEP 2] If the pivot element is equal to zero, goto STEP 7.
- [STEP 3] Control falls through to this point if the pivot is nonzero, in which case divide ROW by pivot.
- [STEP 4] DET = products of pivot elements.
- [STEP 5] Form zeros in the remaining rows below the current pivot element.
- [STEP 6] If all the pivots have been found, return to caller, else goto step 1.
- [STEP 7] If all the remaining columns have been added, return to caller with a DET = 0, else fall through to STEP 8.
- [STEP 8] Add next column to the column which has the pivot term equal to zero, then goto STEP 1.
- [STEP 9] Pass control back to caller.

Although the algorithm is not directly translatable to any computer language, it does describe the main flow of the routine. Inspection of the code provides the detailed information needed to form a determinant.

Matrix Inverse (mat_inv())

The inverse of a matrix is very useful in solving linear equations because multiplication of a matrix by its inverse forms a unity matrix. The property of a unity matrix states that any matrix multiplied by unity matrix equals the original matrix. These two facts allow you to solve linear equations. (The process is covered later in the article.)

The method for finding the inverse matrix is similar to finding the determinant of a matrix. The matrix is decomposed to form a unity matrix (ones along the major diagonal). While this decomposition is performed, the resulting inverse matrix is stored in the original matrix. An index of where the final elements should be placed is maintained. The index is then used to place the inverse elements in proper locations at the end of the routine. The overall algorithm:

- [STEP 1] LOOP for the number of columns in the input matrix.
- [STEP 2] Find the pivot element.
- [STEP 3] If necessary, interchange rows to put the pivot element on the diagonal (by saving index to the row, and the column of pivot element).
- [STEP 4] DIVIDE the pivot row by the pivot element at the row and the column.
- [STEP 5] Reduce all rows, except the pivot row.
- [STEP 6] Pass control back to top of loop in step 1.
- [STEP 7] Interchange pairs of columns saved in index array (STEP 3), in reverse order, to form proper inverse matrix.
- [STEP 8] Pass control back to caller.

Matrix Copy (mat_cpy())

Although copying a matrix is not very complicated, this routine is included in the matrix tool kit because it will be used very frequently. For example, when the matrix operation involves replacing the original matrix with the resulting matrix, Matrix Copy is necessary because the matrix library is written in such a fashion that operations performed by the subroutine are nondestructive to the input matrices.

Matrix display (pmat())

This routine was written as a debug aid. It displays a square matrix on the console device.

Linear Equations Solutions

After all the matrix routines have been entered and debugged, the routines can be used to solve a set of linear equations. Setting up the operational rules and guidelines is the first step in forming a solutions. Consider the following set of linear equations with m equations and n unknowns:

$$\begin{aligned} \alpha(1,1) x(1) + \alpha(1,2) x(2) &= k(1) \\ \alpha(2,1) x(1) + \alpha(2,2) x(2) &= k(2) \end{aligned}$$

Figure 1

In this example, the A and K terms would have defined values, and the X terms would be the variables yet to be determined. The object would be to find values for x(1) and x(2), which would allow k(1) and k(2) be valid results for the set of equations. The general form for the set of equations in Figure 1:

$$(A)(n \times m) (X)(m \times 1) = (K)(m \times 1)$$

In this form, the term [A] is a set of known coefficients in a matrix of order (n x m). The unknown terms, or variables, are given by the [X] matrix, which has an order of (m x 1). The set of [K] terms includes the known result for each equation and has an order of (m x 1). Once the equations are set up in this form, 5 steps must be taken to find the values of the unknown variables.

Step 1

Multiply both sides of the equations by the inverse of the A matrix. This calculation results in the left side of the equation containing the A matrix multiplied by its inverse, multiplied by the variable matrix. The right side of the equation contains the K matrix multiplied by the inverse of the A matrix.

$$(A)^{-1} (A) (X) = (A)^{-1} (K)$$

Step 2

The next step is to substitute the value of the A matrix multiplied by its inverse with the unity matrix on the left side of the equations. This results in the following representations.

$$(I) (X) = (A)^{-1} (K)$$

AMIGA HARD DISK BACKUP HARDHAT

Full/Incremental/Directory/Single File backup to microdisks. Option list allows skipping of files by name with wildcards. Catalog file provides display of backed up files by name with size, location and datestamp. Double data compression reduced disk space. Printer interface. Uses CLI or Workbench. Multitasking provides background operation. — \$69.95

AMIGA DISK FILE ORGANIZER ADFO

Having trouble finding that file somewhere in your stack of floppies? Can't find all the copies of a particular file? ADFO maintains a database of directories and filenames from your collection of disks. Fast response inquiries return location and last update information. Printer interface. Uses CLI or Workbench. 512K ram and 2 drives recommended — \$59.95.

AMIGA SPELLING CHECKER SPEL-IT

Uses 40,000 word primary dictionary and optional second dictionary. Add/Delete words to both dictionaries. Includes plurals. Text wordcount totals. Uses CLI or Workbench, Mouse or keyboard. — \$49.95

Include \$3.50 S&H Mastercard/Visa Accepted
Calif. Residents Add 6½% Sales Tax

Westcom Industries

3386 Floyd
Los Angeles, CA 90068 (213) 851-4868
Order phone 1 800 621-0849 Ext. 494

Step 3

The left side of the equation can again be reduced to the X matrix. The fact that any matrix multiplied by the unity matrix is equivalent to the matrix itself causes the reduction. The next form of the solution is shown below.

$$(X) = (A)^{-1} (K)$$

Step 4

The right side of the equation is now reduced by multiplying the known value [K] matrix by the inverse of the [A] matrix. This operation leaves one matrix on each side of the equation of the order (m x 1).

$$(X) = (R)$$

Step 5

The two remaining matrices are now expanded to uncover the value of each of the unknown variables from the original equation set. This expansion, which takes the following form, presents the unknown variable reference and its scalar value which is necessary to satisfy the set of simultaneous equations.

$$x(1) = \text{value}_1$$

(continued)

Listing One

```

/*****
* File:      matlib.c
* Function:   Floating point matrix library
* Date:      Aug 1987
* Author:    Robert W. Ellis
* Copyright 1987 Robert W. Ellis
* Change list:
*****/

/*****
* Cautions:  The allocated size of the matrices used as input
*            to the library routines MUST be the same size
*            as the matrices that are defined in the subroutines.
*            This is a limitations caused by the way in which
*            2 dimensional arrays are addressed in C.
*****/

#include <stdio.h>

/*****
* Local defines
*****/

#define YES 1
#define NO 0

/*****
* Entry points to library
*****/

extern float  mat_det();
extern int    mat_mul();
extern int    mat_inv();
extern int    mat_add();
extern int    mat_sub();
extern int    mat_cpy();
extern int    mat_scl();
extern float  f_abs();
extern int    pmatr();

/*****
* Main test routine for matrix library. This code should be removed
* after testing of library is complete.
*****/

main()
{
    static float a[10][10];
    static float b[10][10];
    static float c[10][10];
    static float rslt;

    printf("Matrix lab\r\n");
    a[0][0] = 4;
    a[0][1] = 2;
    a[0][2] = 8;
    a[1][0] = 12;
    a[1][1] = 8;
    a[1][2] = 16;
    a[2][0] = 16;
    a[2][1] = 32;
    a[2][2] = 16;

    b[0][0] = 3;
    b[0][1] = 1;
    b[0][2] = 1;
    b[1][0] = 2;
    b[1][1] = 1;
    b[1][2] = 3;
    b[2][0] = 1;
    b[2][1] = 2;
    b[2][2] = 1;

    /*****
    * Test case for matrix addition
    *****/

    printf("Matrix addition test\n");
    pmatr(" A matrix",a,2);
    pmatr(" B matrix",b,2);
    mat_add(a,2,2,b,c);
    pmatr(" C result matrix",c,2);
    nextkey();

    /*****
    * Test case for matrix subtraction
    *****/

    printf("Matrix subtraction test\n");
    pmatr(" A matrix",a,2);
    pmatr(" B matrix",b,2);
    mat_sub(a,2,2,b,c);
    pmatr(" C result matrix",c,2);
    nextkey();

    /*****
    * Test of matrix multiplication
    *****/

    printf("Matrix multiplication test\n");
    pmatr(" A matrix",a,2);
    pmatr(" B matrix",b,2);
    mat_mul(a,2,2,b,c);
    pmatr(" C result matrix",c,2);
    nextkey();

    /*****
    * Test of a determinant of a matrix
    *****/

    printf("Determinant of a matrix\n");
    pmatr(" A matrix",a,3);
    mat_cpy(a,3,3,c);
    rslt = mat_det(a,3);
    if( rslt != 640 )
    {
        printf("Answer %f is NOT correct\n",rslt);
    }
    else
    {
        printf("Answer %f IS correct\n",rslt);
    }

    /*****
    * Test of inverse matrix operation
    *****/

    a[0][0] = 1;
    a[0][1] = 2;
    a[0][2] = 3;
    a[1][0] = 4;
    a[1][1] = 2;
    a[1][2] = 1;
    a[2][0] = 1;
    a[2][1] = 1;
    a[2][2] = 1;
    printf("Matrix inverse test\n");
    pmatr(" A matrix",a,3);
    mat_inv(a,3);
    pmatr(" Inverse ",a,3);

    a[0][0] = 1;
    a[0][1] = 1;
    a[0][2] = -4;
    a[1][0] = -3;
    a[1][1] = -2;
    a[1][2] = 11;
    a[2][0] = 2;
    a[2][1] = 1;
    a[2][2] = -6;
    printf("Resulting inverse matrix should be\n");
    pmatr(" Inverse matrix",a,3);

    } /* end of main */
    /***** End of test routines Start of library *****/

    /*****
    * Procedure : mat_det
    * Function  : This procedure will calculate the determinant
    *            of a square matrix.
    * Input     : b - Input matrix
    *            : size - Number of ROWs and COLUMNS in matrix
    *****/

    float  mat_det(b,size)
    float  b[10][10]; /* input matrix */
    int    size;      /* matrix size */
    {
        int i,j,n,k; /* local loop counters */
        float det; /* */
        float tol; /* */
        float pivot;
        float dmul;
        float c;
        int done;
        int ncol;

        det = 1;
        tol = 1e-10;

```



```

for( n=0; n < size; n++)
{
    ncol = n;
    done = NO;

    while( ! done )
    {
        pivot = b[n][n];
        if( n == (size-1) )
        {
            det = det * pivot;
            done = YES;
        }
        else
        {
            if( f_abs(pivot) > tol )
            {
                /*****
                 * subsection A
                 *****/
                for( j=0; j < size; j++)
                {
                    b[n][j] = b[n][j] / pivot;
                }
                k = n + 1;
                for( i = k; i < size; i++ )
                {
                    dmul = b[i][n];
                    for( j = n; j < size; j++ )
                    {
                        c = -dmul * b[n][j];
                        b[i][j] = b[i][j] + c;
                    }
                }
                det = det * pivot;
                done = YES;
            }
            else
            {
                /*****
                 * Subsection B
                 *****/

                ncol += 1;
                for( j = 0; j < size; j++ )
                {
                    b[j][n] = b[j][n] + b[j][ncol];
                }
                if( ncol > size )
                {
                    det = 0;
                }
                if( det == 0 )
                    return( det );

                done = NO;
            }
        }
    } /* end of while */
} /* end of for n */
return( det );
} /* end of mat_det_int */

/*****
* Procedure : mat_mul
* Function : This procedure will multiply two input matrices
* to form a third output matrix.
* Input : a - Input matrix A
* : ira - Number of ROWS in matrix A
* : ica - Number of COLUMNS in matrix A
* : b - Input matrix B
* : icb - Number of COLUMNS in matrix B
* : c - Resulting output matrix
*****/

int mat_mul(a,ira,ica,b,icb,c)
float a[10][10],b[10][10],c[10][10];
int ira,ica,icb;
{
    int i,j,k;

    for( i=0; i < ira; i++)
    {
        for( j=0; j < icb; j++)
        {
            c[i][j] = 0.0;
            for( k=0; k < ica; k++)
            {
                c[i][j] += a[i][k] * b[k][j];
            }
        }
    }
}

```

Reg. \$299.95 **Special Price....** For A Limited Time.... Reg. \$299.95

1 MEG Ram Expansion **AMMEG1™ only \$249.95***

Are you afraid that a low price means low quality, or there's a catch, or you probably have to buy something more, like ram chips?

Our **AMMEG1™** comes in a heavy metal case. It comes with all ram chips installed. There's no catch. Nothing more to buy. Just plug it in and power up. It's that simple.

If you want pass thru, a lot of glitter that you may never use and a price that's \$100.00 to \$300.00 more, then the **AMMEG1™** is not for you.

But, if all you want is to add 1 meg of memory, at the lowest price available, then call or write us today. If you are not satisfied we will cheerfully refund the purchase price.

You have nothing to lose and 1 meg of affordable memory to gain.

- Auto-Configure & Zero Waits
- Fully Assembled & Tested
- 90-Day Parts & Labor Warranty

KLINE-TRONICS

10 Carlisle Court York, PA 17404

Tel. (717)-764-4205

* Plus Shipping & Handling

```

return(1); /* return and signal NO errors */
}
/* end of routine mat_mul */

```

```

/*****
* Procedure : mat_inv
* Function : This procedure will invert the input matrix
*
* Input : a - Input matrix
* : n - Number of ROWS and COLUMNS in matrix
*
* Output : a - Input matrix is altered and becomes
* : the resulting matrix
*****/

```

```

mat_inv(a,n)
float a[10][10];
int n; /* size of input square matrix */
{
    static int ipiv[20];
    static int indxr[20],indx[20];
    int i,j,l,k;
    int irow,icol;
    static float dum;
    static float pivinv;

```

```

float big;

for( j = 0; j < n; j++)
{
    ipiv[j] = 0;
}

for( i = 0; i < n; i++)
{
    big = 0;
    for( j = 0; j < n; j++)
    {
        if( ipiv[j] != 1 )
        {
            for( k = 0; k < n; k++)
            {

```

(continued)


```

        if( ipiv[k] == 0 )
        {
            if( f_abs(a[j][k]) > big )
            {
                big = f_abs(a[j][k]);
                irow = j;
                icol = k;
            }
        }
        else if( ipiv[k] > 1 )
        {
            printf("singular matrix\n");
            return(0);
        }
    }
}

ipiv[icol] = ipiv[icol] + 1;

if( irow != icol )
{
    for( l = 0; l < n; l++)
    {
        dum = a[irow][l];
        a[irow][l] = a[icol][l];
        a[icol][l] = dum;
    }
}

indxr[i] = irow;
indxc[i] = icol;
if( a[icol][icol] == 0 )
{
    printf("singular matrix\n");
    return(0);
}
pivinv = 1 / a[icol][icol];
a[icol][icol] = 1;

for( l = 0; l < n; l++)
{
    a[icol][l] = a[icol][l] * pivinv;
}

for( ll = 0; ll < n; ll++)
{
    if( ll != icol )
    {
        dum = a[ll][icol];
        a[ll][icol] = 0;
        for( l = 0; l < n; l++)
        {
            a[ll][l] = a[ll][l] - a[icol][l] * dum;
        }
    }
}

for( l = n-1; l >= 0; l--)
{
    if( indxr[l] != indxc[l] )
    {
        for( k = 0; k < n; k++)
        {
            dum = a[k][indxr[l]];
            a[k][indxr[l]] = a[k][indxc[l]];
            a[k][indxc[l]] = dum;
        }
    }
}

return(1); /* return a complete NO error condition */
}
/* end of procedure mat_inv */

/*****
* Procedure : mat_add()
* Function : This procedure will add two input matrices and
*           : form a third output matrix
* Input    : a - Input matrix A
*           : ira - Number of ROWS in matrix A
*           : ica - Number of COLUMNS in matrix A
*           : b - Input matrix B
*           : c - Output matrix C
*****/

int mat_add(a,ira,ica,b,c)
float a[10][10];
int ira;
int ica;
float b[10][10];
float c[10][10];
{
    int i,j;

```

```

    for( i = 0; i < ica; i++)
    {
        for( j = 0; j < ira; j++)
        {
            c[i][j] = a[i][j] + b[i][j];
        }
    }

    return(1);

/* return a complete NO error condition */
} /* end of procedure mat_add */

/*****
* Procedure : mat_sub()
* Function : This procedure will subtract two input matrices and
*           : form a third output matrix
* Input    : a - Input matrix A
*           : ira - Number of ROWS in matrix A
*           : ica - Number of COLUMNS in matrix A
*           : b - Input matrix B
*           : c - Output matrix C
*****/

int mat_sub(a,ira,ica,b,c)
float a[10][10];
int ira;
int ica;
float b[10][10];
float c[10][10];
{
    int i,j;

    for( i = 0; i < ica; i++)
    {
        for( j = 0; j < ira; j++)
        {
            c[i][j] = a[i][j] - b[i][j];
        }
    }

    return(1);

/* return a complete NO error condition */
} /* end of procedure mat_add */

/*****
* Procedure : mat_cpy
* Function : This procedure will make a copy of a matrix
* Input    : a - Input matrix A
*           : ira - Number of ROWS in matrix A
*           : ica - Number of COLUMNS in matrix A
*           : b - Output matrix B
*****/

int mat_cpy(a,ira,ica,b)
float a[10][10];
int ira;
int ica;
float b[10][10];
{
    int i,j; /* local loop counters */

    for( i = 0; i < ica; i++)
    {
        for( j = 0; j < ira; j++)
        {
            b[i][j] = a[i][j];
        }
    }

    return(1); /* return a complete NO error condition */
} /* end of procedure mat_cpy */

/*****
* Procedure : mat_scl
* Function : This procedure will multiply a matrix by
*           : a scalar
* Input    : a - Input/Output matrix A
*           : ira - Number of ROWS in matrix A
*           : ica - Number of COLUMNS in matrix A
*           : scalar - value to multiply elements by
*****/

```

(continued on page 98)

If You are Searching for a
Monthly
Resource to
the Commodore
AMIGA™....



Amazing
COMPUTING

Wake up
and Smell the Coffee!!


```

int mat_scl(a,ira,ica,scalar)
float a[10][10];
int ira;
int ica;
float scalar;
{
    int i,j; /* local loop counters */

    for( i = 0; i < ica; i++)
    {
        for( j = 0; j < ira; j++)
        {
            a[i][j] = a[i][j] * scalar;
        }
    }

    return(1); /* return a complete NO error condition */
} /* end of procedure mat_scl */

/*****
*
* Support and Debug routines
*
*****/

/*****
* Procedure : f_abs
* Function : This procedure will the absolute value of
*           a floating point number.
*
* Input : val - Pointer to input signed floating
*         point number
*
*****/

float f_abs(val)
float *val;
{
    return( (*val > 0) ? *val : -*val );
}

/*****
* Procedure : pmats
* Function : This procedure will display a square matrix
*           of floating point numbers
*
* Input : s - Pointer to header string
*        a - Input matrix A
*        n - Number of rows and columns
*
*****/

pmats(s,a,n)
char *s;
float a[10][10];
int n;
{
    int i,j;
    static float temp;

    printf("Pmats - %s\n",s);

    for( i = 0; i < n; i++)
    {
        for( j = 0; j < n; j++)
        {
            temp= a[i][j];
            printf("%f ",temp);
        }
        printf("\n");
    }
} /* end of procedure pmats */

/* end of file matlib.c */

nextkey()
{
    printf("Hit any key to continue\n");
    kbhit();
    getch();
}

```

Listing Two

```

\ File:      matlib
\ Function:   Floating point matrix library
\ Language:   AmigaBASIC V1.2
\ Date:      Aug 1987
\ Author:    Robert W. Ellis
\ Copyright 1987 Robert W. Ellis

```

\ Change list:

\ Global data allocation

```

DEFDBL a,b,c,h,r,s
DIM a(10,10)
DIM b(10,10)
DIM c(10,10)
DEFINT i,n

```

\ Subroutine shared variables

```

\ subroutine mat_inv
DIM ipiv(20)
DIM indxr(20),indxc(20)

```

\ Main program and test area

```

PRINT "matlib test program"
a(0,0)=4
a(0,1)=2
a(0,2)=8
a(1,0)=12
a(1,1)=8
a(1,2)=16
a(2,0)=16
a(2,1)=32
a(2,2)=16

```

```

b(0,0)=3
b(1,0)=2
b(2,0)=1
b(0,1)=1
b(1,1)=1
b(2,1)=2
b(0,2)=1
b(1,2)=3
b(2,2)=1

```

```

*****
\ Test case for matrix addition
*****
PRINT "addition test"
PRINT " A matrix"
CALL pmats(2,a())
PRINT " B matrix"
CALL pmats(2,b())
CALL matadd(a(),2,2,b(),c())
PRINT " C result matrix"
CALL pmats(2,c())
PRINT "hit return to continue"
INPUT ch$

```

```

*****
\ Test case for matrix subtraction
*****
PRINT "subtraction test"
PRINT " A matrix"
CALL pmats(2,a())
PRINT " B matrix"
CALL pmats(2,b())
CALL matsub(a(),2,2,b(),c())
PRINT " C result matrix"
CALL pmats(2,c())
PRINT "hit return to continue"
INPUT ch$

```

```

*****
\ Test case for matrix multiplication
*****
PRINT "mult test"
PRINT " A matrix"
CALL pmats(2,a())
PRINT " B matrix"
CALL pmats(2,b())
CALL matmul(2,2,2,a(),b(),c())
PRINT " C matrix"
CALL pmats(2,c())
PRINT "hit return to continue"
INPUT ch$

```



```

*****
' Test case for determinant of a matrix
*****
PRINT "det test"
PRINT " A matrix "
CALL pmats(3,a())
CALL matcpy(a(),3,3,c())
CALL matdet(3,r,c())
PRINT " Determinant result r =";r
IF ( r = 640 ) THEN
  PRINT " answer is correct "
ELSE
  PRINT " answer is NOT correct"
END IF
PRINT "hit return to continue"
INPUT ch$

a(0,0)=1
a(0,1)=2
a(0,2)=3
a(1,0)=4
a(1,1)=2
a(1,2)=1
a(2,0)=1
a(2,1)=1
a(2,2)=1

*****
' Test case for matrix inversion
*****
PRINT "Inverse matrix"
PRINT " A matrix"
CALL pmats(3,a())
CALL matinv(3,a())
PRINT "result matrix"
CALL pmats(3,a())
a(0,0)=1
a(0,1)=1
a(0,2)=-4
a(1,0)=-3
a(1,1)=-2
a(1,2)=11
a(2,0)=2
a(2,1)=1
a(2,2)=-6
PRINT "should match the following"
CALL pmats(3,a())

PRINT "hit return to continue"
INPUT ch$

*****
' Test case for scalar multiplication
*****
PRINT "Scalar multiplication by 3"
PRINT " B matrix"
CALL pmats(2,b())
CALL matscl(b(),2,2,3!)
PRINT " B result matrix"
CALL pmats(2,b())
PRINT "hit return to continue"
INPUT ch$

PRINT "end of main"
END
' end of main program

*****
' Subroutine : mat_inv
' Function : invert a matrix
' Inputs :
*****
SUB matinv(n%,a(2)) STATIC

SHARED ipiv()
SHARED indxr()
SHARED indxc()

PRINT "mat_inv"
n= n-1
FOR j=0 TO n
  ipiv(j) = 0
NEXT j
FOR i=0 TO n
  big=0!
  FOR j=0 TO n
    IF ( ipiv(j) > 1 ) THEN
      FOR k=0 TO n
        IF ( ipiv(k) = 0 ) THEN
          big= ABS(a(j,k))

```

```

          irow = j
          icol = k
        ELSEIF ( ipiv(k) > 1 ) THEN
          PRINT "singular matrix"
          RETURN
        END IF
      NEXT k
    END IF
  NEXT j

  ipiv(icol) = ipiv(icol)+1
  IF ( irow > icol ) THEN
    FOR l=0 TO n
      dum=a(irow,l)
      a(irow,l) = a(icol,l)
      a(icol,l) = dum
    NEXT l
  END IF

  indxr(i) = irow
  indxc(i) = icol

  IF ( a(icol,icol) = 0! ) THEN
    PRINT "singular matrix"
    RETURN
  END IF

  pivinv = 1/a(icol,icol)
  a(icol,icol)=1

  FOR l=0 TO n
    a(icol,l) = a(icol,l) * pivinv
  NEXT l

  FOR ll=0 TO n
    IF ( ll > icol ) THEN
      dum = a(ll,icol)
      a(ll,icol) = 0
      FOR l=0 TO n
        a(ll,l) = a(ll,l) - a(icol,l) * dum
      NEXT l
    END IF
  NEXT ll
NEXT i

FOR l= n TO 0 STEP -1
  IF ( indxr(l) > indxc(l) ) THEN
    FOR k=0 TO n
      dum = a(k,indxr(l))
      a(k,indxr(l)) = a(k,indxc(l))
      a(k,indxc(l)) = dum
    NEXT k
  END IF
NEXT l

END SUB
***** end of subroutine mat_inv *****

*****
' Subroutine : mat_mul
' Function : multiply two matrices
' Inputs : a - input matrix A
'          ira- number of rows in A
'          irc- number of columns in A
'          b - input matrix B
'          icb- number of columns in B
'          c - output matrix C
*****

SUB matmul(nra%,nca%,ncb%,a(2),b(2),c(2)) STATIC

nra = nra - 1
ncb = ncb - 1
nca = nca - 1

FOR i=0 TO nra
  FOR j=0 TO ncb
    c(i,j) = 0!
    FOR k=0 TO nca
      c(i,j) = c(i,j) + (a(i,k) * b(k,j))
    NEXT k
  NEXT j
NEXT i
END SUB

' end of subroutine mat_mul

*****
' Subroutine : mat_det
' Function : multiply two matrices
'

```

(continued)


```

` Inputs      : b      - input matrix B
`             : nsize - size of square matrix
` Outputs     : rdet   - determinant scalar
`*****

SUB matdet(nsize%,rdet,b(2)) STATIC
nsize = nsize - 1
rdet = 1!
tol = 1E-10

FOR n=0 TO nsize
  ncol = n
  done = 0

  WHILE( done = 0 )
    pivot = b(n,n)
    IF( n = (nsize) ) THEN
      rdet = rdet * pivot
      done = 1
    ELSE
      IF( ABS( pivot ) > tol ) THEN
        FOR j=0 TO nsize
          b(n,j) = b(n,j) / pivot
        NEXT j
        k = n + 1

        FOR i=k TO nsize
          dmul = b(i,n)
          FOR j=n TO nsize
            temp = -dmul * b(n,j)
            b(i,j) = b(i,j) + temp
          NEXT j
        NEXT i

        rdet = rdet * pivot
        done = 1
      ELSE
        ncol = ncol + 1
        FOR j=0 TO nsize
          b(j,n) = b(j,n) + b(j,ncol)
        NEXT j
        IF( ncol > nsize ) THEN
          rdet = 0
        END IF

        IF( rdet = 0 ) THEN
          GOTO detend
        END IF
        done = 0
      END IF
    END IF
  WEND
NEXT n
detend:

END SUB
` ***** end of subroutine mat_det *****

`*****
` Subroutine : mat_add
` Function   : add two matrices
`
` Inputs    : a      - Input matrix A
`            : nra    - Number of ROWs in matrix A
`            : nca    - Number of COLUMNS in matrix A
`            : b      - Input matrix B
` Outputs   : c      - Output matrix C
`*****

SUB matadd(a(2),nra%,nca%,b(2),c(2)) STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    c(i,j) = a(i,j) + b(i,j)
  NEXT j
NEXT i

END SUB
` ***** end of subroutine mat_add *****

`*****
` Subroutine : mat_sub
` Function   : subtract two matrices
`
` Inputs    : a      - Input matrix A
`            : nra    - Number of ROWs in matrix A
`            : nca    - Number of COLUMNS in matrix A

```

```

`             : b      - Input matrix B
` Outputs   : c      - Output matrix C
`*****

SUB matsub(a(2),nra%,nca%,b(2),c(2)) STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    c(i,j) = a(i,j) - b(i,j)
  NEXT j
NEXT i

END SUB
` ***** end of subroutine mat_sub *****

`*****
` Subroutine : mat_cpy
` Function   : copy a matrix
`
` Inputs    : a      - Input matrix A
`            : nra    - Number of ROWs in matrix A
`            : nca    - Number of COLUMNS in matrix A
` Outputs   : b      - Output matrix b
`*****

SUB matcpy(a(2),nra%,nca%,b(2)) STATIC
nra = nra - 1
nca = nca - 1

FOR i=0 TO nca
  FOR j=0 TO nra
    b(i,j) = a(i,j)
  NEXT j
NEXT i

END SUB
` ***** end of subroutine mat_cpy *****

`*****
` Subroutine : mat_scl
` Function   : multiply a matrix by a scalar
`
` Inputs    : a      - Input matrix A
`            : nra    - Number of ROWs in matrix A
`            : nca    - Number of COLUMNS in matrix A
`            : scl    - scalar value
` Outputs   : a      - Output matrix A
`*****

SUB matscl(a(2),nra%,nca%,s!) STATIC
nra = nra - 1
nca = nca - 1

PRINT "nra ";nra;" nca";nca;" scl";s!
FOR i=0 TO nca
  FOR j=0 TO nra
    a(i,j) = a(i,j) * s!
  NEXT j
NEXT i

END SUB
` ***** end of subroutine mat_scl *****

`*****
`* Start of support and debug routines
`*****

`*****
` Subroutine : pmat
` Function   : print a square matrix
`*****

SUB pmat(n%,a(2)) STATIC
n = n-1

FOR i=0 TO n
  PRINT " | " ;
  FOR j=0 TO n
    PRINT USING " #####.#####";a(i,j);
  NEXT j
  PRINT " |"
NEXT i
END SUB

```

•AC•

by Steve Faiwizewski

Modula-2 Programming

on the Amiga™

Calc Continued...

[The following source is the remainder of IMPLEMENTATION
MODULE Interpreter, which appeared in AC V3.1. Also listed are
the remaining modules to the Calc program, introduced in AC V2.12
-Ed]

```
'Q' :
  x := Evaluate(root^.lson, suc);
  IF x >= 0.0 THEN RETURN sqrt(x)
ELSE
  HandleError(InvalidValueForSqrtError, 0);
  what := oops;
END

'V' : RETURN(SubstituteVar(root))
'' :
(* Because of the way power is implemented in TDI's *)
(* MathLib0, we have to calculate it ourselves if x *)
(* is negative. We take advantage of the fact that *)
(* x^p = exp(ln(x)*p) (Thanks, Art!) *)

  x := Evaluate(root^.lson, suc);
  IF x = 0.0 THEN
    RETURN 0.0
  ELSE
    p := Evaluate(root^.rson, suc);
    IF x > 0.0 THEN
      RETURN exp(ln(x)*p)
    ELSE (* x is negative *)
      x := ABS(x);
      IF entier(p) MOD 2 = 0 THEN
        RETURN exp(ln(x)*p)
      ELSE
        RETURN -(exp(ln(x)*p))
      END
    END (* x is negative *)
  END
END

ELSE
  WriteLine('*** Evaluate: Program Error ***');
  suc := FALSE;
END (* CASE *)
ELSE
  IF (root=NIL) THEN
    WriteLine('Evaluate: Program Error! root is NIL');
  END;
  suc := FALSE;
END;
IF what <> oops THEN
  WriteLine('Unknown Problem in Evaluation -');
  what := oops;
END;
RETURN(0.0)
END Evaluate;

PROCEDURE InitStack;
BEGIN
  (* initialize stacks and stack pointers *)
  tstack:=0;
  ostack:=1;

  (* next expected token is either a number or "(" *)
  (* so whatwas must be set to operator. *)
  whatwas := operator;
  what := operator;
END InitStack;
```

```
PROCEDURE FirstInit;
VAR
  i : CARDINAL;
  t1: nodep;
BEGIN
  (* initialize operator stack *)
  NEW(t1);
  t1^.lson := NIL;
  t1^.rson := NIL;
  t1^.oper:='*';
  operstack[1]:=t1;
  ostack:=1;

  FOR i:=0 TO 26 DO
    valdef[i] := FALSE;
  END;
  FOR i:= 1 TO maxtree DO
    treestack[i] := NIL;
  END;
  FOR i:=2 TO maxoper DO
    operstack[i] := NIL;
  END;
END FirstInit;

PROCEDURE FinishUp;
BEGIN
  DestroyHeap;
END FinishUp;

BEGIN
  FirstInit;
END Interpreter.

MODULE Calc;
(* ***** *)
(* (c) Copyright 1986, 1987 by *)
(* Steve Faiwizewski & Richie Bielak *)
(* *)
(* For non-commercial, non-profit use only. *)
(* ***** *)

(*$A*) (* Change all JSRs to BSRs in all modules *)
(*$S*)
(*$T*)

FROM InOut IMPORT WriteString, WriteCard, WriteLn,
  Write, Read, OpenInputOutputFile,
  CloseInputOutput;
FROM MyInOut IMPORT ReadString;
FROM MyRealOut IMPORT WriteReal;
FROM Interpreter0 IMPORT treestack, WriteLine, Trim;
FROM Interpreter IMPORT CharSet, Evaluate, GrowTree,
  InitStack, tstack, what, symtable,
  valdef, tokentype, ErrorType,
  HandleError, exptype, FinishUp;
FROM WBStart IMPORT LaunchedFromWB, ReturnToWB;
FROM Strings IMPORT Length, Concat;
FROM CommandLine IMPORT CLStrings, GetCL;
FROM DOSProcessHandler
  IMPORT Exit;
FROM AMIGAX IMPORT ErrorProcessor;
(* IMPORT Trapper; *)
```

(continued)


```

CONST
  MaxArgs = 10;
  LF = 12C;
  FF = 14C;
  CR = 15C;

VAR
  linesize,
  point,
  argc,
  offset,
  digs,i
  : CARDINAL;
  line
  : CLStrings;
  exptype
  : exptype;
  argv
  : ARRAY[0..MaxArgs] OF CLStrings;
  result
  : REAL;
  suc,
  FromCommandLine,
  done
  : BOOLEAN;

PROCEDURE dohelp;
BEGIN (* DO HELP *)
  Write(FF);
  WriteLn(
    'This CALCulator program lets you do various calculations.'):
  WriteLn('Enter expressions as you would write them.'):
  WriteLn('Supported functions and operators are:');
  WriteLn(' +, -, *, /, ^ (power of), ');
  WriteLn(' ABS, SIN, COS, TAN, LOG, LN, EXP, SQRT.'):
  WriteLn;
  WriteLn(
    'In addition you can use up to 26 variables (A thru Z)'):
  WriteLn('To store values, for example:');
  WriteLn('      You enter   -> a=2*4');
  WriteLn('      You enter   -> a');
  WriteLn('      Output      -> 8.00');
  WriteLn('      You enter   -> b=a*2');
  WriteLn('      You enter   -> b');
  WriteLn('      Output      -> 16.00');
  WriteLn;
  WriteLn(
    'You can use variables in the expressions too.'):
  WriteLn(
    '(for example b=23*a). To change the number of decimal'):
  WriteLn(
    'places displayed use /Dn where n is the desired number of'):
  WriteLn(
    'decimal places. Valid values are 1 through 7.'):
  WriteLn;
  WriteLn('Type /A to find out more about Calc');
  WriteLn('Type /E to Exit.'):
END dohelp;

PROCEDURE HelpSwitch;
BEGIN
  WriteLn('One of the following : ');
  WriteLn('      /Dn : for n decimal places');
  WriteLn('      /H : for help');
  WriteLn('      /E : to exit');
END HelpSwitch;

PROCEDURE ErrorHandler(err: ErrorType; level: CARDINAL);
BEGIN
  WriteString('*** Error : ');
  CASE err OF
    MissingValueError:
      WriteString('Undefined Variable');
    MissingOperatorError:
      WriteString('Missing Operator');
    MissingOperandError:
      WriteString('Missing Operand');
    MissingOpenParenError:
      WriteString('Missing Open Parenthesis');
    MissingCloseParenError:
      WriteString('Missing Close Parenthesis');
    IllegalValueForTanError:
      WriteString('Illegal Value for Function Tan');
    IllegalValueForLogError:
      WriteString('Illegal Value for Function Log');
    IllegalValueForSqrtError:
      WriteString('Illegal Value for Function Sqrt');
    IllegalValueForExpError:
      WriteString('Illegal Value for Function Exp');
    DivideByZeroError:
      WriteString('Divide by zero error!');
    GeneralError:
      WriteString('General type');
  ELSE
    WriteString('Unknown type')
  END;
  WriteLn(' (Level = '); WriteCard(level,1);
  WriteLn(') ***');
END ErrorHandler;

```

```

PROCEDURE PrepareToExit;
BEGIN
  FinishUp;
  IF NOT FromCommandLine THEN CloseInputOutput END;
  IF LaunchedFromWB THEN ReturnToWB END;
END PrepareToExit;

PROCEDURE ErrorTrapper;
(* If we get here, PANIC! *)
(* Display an error msg, then close *)
(* everything and quit. *)
VAR
  dummy : ARRAY[0..1] OF CHAR;
BEGIN
  WriteLn('*** Runtime Error!!! Gotta Quit!! ***');
  WriteString('Save all other work, then press RETURN: ');
  ReadString(dummy);
  PrepareToExit;
  Exit(99);
END ErrorTrapper;

PROCEDURE GetInput;
BEGIN
  REPEAT
    IF FromCommandLine THEN
      line := argv[0];
      done := TRUE;
    ELSE
      WriteString('CALC');
      WriteCard(digs,1);
      Write('>');
      ReadString(line);
    END;
    Trim(line);
    linesize:=Length(line);
  UNTIL linesize>0;
END GetInput;

PROCEDURE UpperCaseLine(VAR line : ARRAY OF CHAR; size : CARDINAL);
VAR
  i : CARDINAL;
BEGIN
  FOR i := 0 TO size-1 DO
    line[i] := CAP(line[i]);
  END;
END UpperCaseLine;

PROCEDURE ProcessSwitches;
BEGIN
  IF line[1] = 'H' THEN
    dohelp
  ELSIF line[1] = 'E' THEN
    done := TRUE
  ELSIF line[1] = '2' THEN
    HelpSwitch
  ELSIF (line[1] = 'D') AND
    (line[2] IN CharSet{'1'..'7'}) THEN
    digs:=ORD(line[2])-ORD('0')
  ELSE
    WriteLn('*** Invalid Switch ***');
  END
END ProcessSwitches;

PROCEDURE DeterminExpressionType(i : CARDINAL): exptype;
BEGIN
  IF line[0] IN CharSet{'A'..'Z'} THEN
    i := 0;
    (* search for an equal sign *)
    WHILE (line[i] <> '=') AND (i < linesize-1) DO
      INC(i);
    END;
    IF line[i] = '=' THEN
      RETURN assg
    END;
  END; (* IF *)
  RETURN expr
END DeterminExpressionType;

PROCEDURE ProcessAssignment(offset : CARDINAL);
VAR
  i : CARDINAL;
BEGIN
  point := offset + 1;
  GrowTree(line,point);
  IF what <> oops THEN
    i := ORD(line[0]) - ORD('A');
    IF tstack > 0 THEN
      suc := TRUE;
      symbtable[i]:=Evaluate(treestack[tstack],suc);
      IF suc THEN
        valdef[i] := TRUE;
      END;
    ELSE

```



```

        WriteLine('*** Invalid Assignment ***')
    END
END
END ProcessAssignment;

PROCEDURE ProcessExpression;
BEGIN
    point := 0;
    GrowTree(line,point);
    IF what <> oops THEN
        suc:=TRUE;
        result := Evaluate(treestack[tstack],suc);
        IF (what<>oops) AND suc THEN
            WriteReal(result,digs);
            WriteLn;
        END
    END
END ProcessExpression;

PROCEDURE ProcessInput;
BEGIN
    UpperCaseLine(line,linesize);
    IF linesize > 0 THEN
        IF line[0]='?' THEN
            dohelp
        ELSEIF line[0] = '/' THEN
            ProcessSwitches;
        ELSE
            exptype := DetermineExpressionType(offset);
            (* do what has to be done *)
            CASE exptype OF
                assg : ProcessAssignment(offset) |
                expr : ProcessExpression
            END;
        END; (* ELSE *)
    END; (* IF *)
END ProcessInput;

BEGIN (* Main Line *)
    done := FALSE;
    HandleError := ErrorHandler;
    ErrorProcessor := ErrorTrapper;
    FromCommandLine := FALSE;
    IF NOT LaunchedFromWB THEN
        IF NOT GetCL(argc,argv) THEN
            WriteLine('Something is wrong with the Command Line!');
        END;
        FromCommandLine := argc > 0;
    END;
    IF NOT FromCommandLine THEN
        OpenInputOutputFile('CON:0/0/640/200/ACalc V1.1b');
        WriteLine(
' (c) Copyright 1986 by Steve Faiwizewski & Richie Bielak');
        WriteLine('Enter ? for Help.');
```

[Included here is MODULE ModemDemo which was accidentally omitted from Steve's column in AC 2.11 - Ed]

MODULE ModemDemo;

```

(* * * * * *)
(* A simple program to demonstrate serial *)
(* port I/O. *)
(* *)
(* This program may be freely distributed *)
(* For non-commercial use only. *)
(* *)
(* Created by Steve Faiwizewski, Sep 87. *)
(* PLINK : THE INTERN *)
(* BIX : theintern *)
(* CIS : 74106,425 *)
(* * * * * *)
```

```

FROM Serial      IMPORT OpenSer, CloseSer, SerRead, SerWrite,
                  QuerySer;

FROM MyRawInOut  IMPORT MyOpen, MyClose, MyRead, MyWrite,
                  MyWriteString, MyWriteLn, MoreCharsComing;

FROM Terminal    IMPORT WriteString, WriteLn;
```

```

FROM SYSTEM      IMPORT BYTE;
FROM AMIGAX      IMPORT ErrorProcessor;
IMPORT Trapper;
```

```

CONST
    TimeInterval = 50;
    CSI = 233C;
```

```

VAR
    OldErrorProcessor : PROC;
```

```

PROCEDURE ShutDown;
BEGIN
    MyClose;
    CloseSer;
END ShutDown;
```

```

PROCEDURE ErrorTrapper;
(* Close our window, and call TDI's error handler *)
BEGIN
    ShutDown;
    OldErrorProcessor;
END ErrorTrapper;
```

```

PROCEDURE CheckForExit(): BOOLEAN;
(* See if F10 was pressed *)
VAR
    c2 : ARRAY [0..1] OF CHAR;
BEGIN
    MyRead(c2[0]);
    MyRead(c2[1]);
    IF (c2[0] = '9') AND (c2[1] = '~') THEN
        RETURN TRUE
    ELSE
        SerWrite(c2,2)
    END;
    RETURN FALSE
END CheckForExit;
```

```

PROCEDURE LoopAround;
CONST
    BufferSize = 256;
VAR
    c : CHAR;
    SerialCharsWaiting : LONGCARD;
    Buffer : ARRAY[0..BufferSize - 1] OF CHAR;
    i : CARDINAL;
    Stop : BOOLEAN;
BEGIN
    Stop := FALSE;
    REPEAT
        WHILE MoreCharsComing(TimeInterval) DO
            MyRead(c);
            IF c = CSI THEN
                Stop := CheckForExit()
            ELSE
                SerWrite(c,1);
            END
        END;
        SerialCharsWaiting := QuerySer();
        IF SerialCharsWaiting > 0 THEN
            SerRead(Buffer,SerialCharsWaiting);
            IF SerialCharsWaiting < BufferSize THEN
                Buffer[CARDINAL(SerialCharsWaiting)] := 0C
            END;
            MyWriteString(Buffer)
        END
    UNTIL Stop;
END LoopAround;
```

```

BEGIN
    (* open serial port at 1200 baud, 7 data bits, 1 stop bit *)
    OpenSer(1200,BYTE(1),BYTE(7),0C,0C,0C,01C);
    MyOpen('RAW:0/0/640/200/TDI M2 Serial Port Demo. Press F10 to
Exit.');
```

•AC•

68000 Assembler Language Programming

on the Amiga™

ASSEMGR.ASM — Part II

[The following source is the remainder of the ASSEMGR.asm graphics routine from AC V3.1. Chris will be back next month with more 68000 Assembler Language Programming. — Ed]

```

move.l #3,d0          * color # 3, set to R=0, G=0, B=15
move.l #0,d1
move.l #0,d2
move.l #15,d3
SYS      SetRGB4 (a6)

move.l graphicsbase,a6
lea.l rastport,a1
move.b RP_JAM1,d0      * JAM1 drawing mode (see rastport.i)
SYS      SetDrMd (a6)   * Set the drawing mode

move.l graphicsbase,a6
lea.l rastport,a1
move.b #0,d1
SYS      SetRast (a6)   * Set the entire screen to color # 0

rts

*****
*          Draw something on the screen          *
*****

drawsomething:

move.l graphicsbase,a6
lea.l rastport,a1
move.l #1,d0
SYS      SetAPen(a6)   * Set the foreground pen to color #1

move.l graphicsbase,a6
lea.l rastport,a1
move.l #10,d0
move.l #12,d1
SYS      Move(a6)      * move the drawing pen to (10,12)

move.l graphicsbase,a6
lea.l rastport,a1
move.l #320,d0
move.l #200,d1
SYS      Draw(a6)      * draw to (320,200)

move.l graphicsbase,a6
lea.l rastport,a1
move.l #2,d0
SYS      SetAPen(a6)   * Set the foreground pen to color #2

move.l graphicsbase,a6
lea.l rastport,a1
move.l #50,d0          * Fill a rectangle from (50,35) to (70,60)
move.l #35,d1
move.l #70,d2
move.l #60,d3
SYS      RectFill(a6)

move.l graphicsbase,a6
lea.l rastport,a1
move.l #3,d0
SYS      SetAPen(a6)   * Set the foreground pen to color #3

move.l graphicsbase,a6
lea.l rastport,a1
move.l #300,d0
move.l #40,d1
SYS      Move(a6)      * move the drawing pen to (300,40)

move.l graphicsbase,a6
lea.l rastport,a1
move.l #3,d0
move.l #190,d1
SYS      Draw(a6)      * draw to (3,190)

rts

```

```

*****
*          END OF PROGRAM          *
*****

```

```

*****
*          Library Definitions          *
*****

```

```

graphics    dc.b 'graphics.library',0
            cnop 0,2

```

```

graphicsbase ds.l 1      * base address of the graphics library

```

```

savesp      ds.l 1      * save stack pointer

```

```

*****
*          Viewport definitions          *
*****

```

```

viewport    ds.l 1      * Next ViewPort in a linked list (none)
            ds.l 1      * ColorMap pointer (none)
            ds.l 1      * Dspins (none)
            ds.l 1      * Sprins (none)
            ds.l 1      * ClrIns (none)
            ds.l 1      * UCopIns (none)
vpwidth     ds.w 1      * Width
vpheight    ds.w 1      * Height
vpdxoffset   ds.w 1      * DxOffset
vpdyoffset   ds.w 1      * DyOffset
vpmodes     ds.w 1      * Modes (just 320x200)
            ds.w 1      * reserved for system use (empty)
vprasinfo   ds.l 1      * Pointer to RasInfo structure

```

```

*****
*          View definitions          *
*****

```

```

view        ds.l 1      * Pointer to viewport
            ds.l 1      * LOFCprList (none)
            ds.l 1      * SHFCprList (none)
dxoffset    ds.w 1      * DxOffset
dyoffset    ds.w 1      * DyOffset
vmodes      ds.w 1      * Modes (just 320x200)

```

```

*****
*          BitMap definitions          *
*****

```

```

bitmap      ds.w 1      * bytes per row (40)
bmrows      ds.w 1      * rows
            ds.b 1      * Flags (none)
bmdepth     ds.b 1      * depth
            ds.w 1      * Padding space
plane1      ds.l 1      * Plane 1 pointer
plane2      ds.l 1      * Plane 2 pointer
plane3      ds.l 1      * Plane 3 pointer
plane4      ds.l 1      * Plane 4 pointer
plane5      ds.l 1      * Plane 5 pointer
plane6      ds.l 1      * Plane 6 pointer
plane7      ds.l 1      * Plane 7 pointer
plane8      ds.l 1      * Plane 8 pointer

```

```

*****
*          RasInfo definitions          *
*****

```

```

rasinfo     ds.l 1      * Next rasinfo structure (none)
ribitmap    ds.l 1      * Pointer to bitmap structure
rrixoffset   ds.w 1      * RxOffset
rixyoffset   ds.w 1      * RyOffset

```

```

*****
*          RastPort definitions          *
*****

```

```

rastport    ds.l 1      * Layer
rpbitmap    ds.l 1      * pointer to bitmap
            dcb.l 17,0  * clear the unused portion

```

end

```

*****
*          End Of Program!!!          *
*****

```


The AMICUS & Fred Fish

Public Domain Software Library

This software is collected from user groups and electronic bulletin boards around the nation. Each Amicus disk is nearly full, and is fully accessible from the Workbench. If source code is provided for any program, then the executable version is also present. This means that you don't need the C compiler to run these programs. An exception is granted for those programs only of use to people who own a C compiler.

The Fred Fish disk are collected by Mr. Fred Fish, a good and active friend of the Amiga.

Note: Each description line below may include something like 'S-O-E-D', which stands for 'source, object file, executable and documentation'. Any combination of these letters indicates what forms of the program are present. Basic programs are presented entirely in source code format.

<p>AMICUS Disk 1</p> <p>Basic programs:</p> <p>3DSolids 3d solids modeling prog, w/sample data files</p> <p>Blocks draws blocks</p> <p>Cubes draws cubes</p> <p>Durer draws pictures in the style of Durer</p> <p>FScape draws fractal landscapes</p> <p>Hidden 3D drawing program, w/hidden line removal</p> <p>JPad simple paint program</p> <p>Optical draw several optical illusions</p> <p>PaintBox simple paint program</p> <p>Shuttle draws the Shuttle in 3d wireframe</p> <p>SpaceArt graphics demo</p> <p>Speaker speech utility</p> <p>Sphere draws spheres</p> <p>Spiral draws color spirals</p> <p>ThreeDoe 3d function plots</p> <p>Topography artificial topography</p> <p>Wheels draws circle graphics</p> <p>Xenos draws fractal planet/landscapes</p>	<p>AMICUS Disk 2</p> <p>C programs:</p> <p>allib AmigaDOS object library manager, S-E</p> <p>ar text file archive program, S-E</p> <p>auto-chops auto-chops executables files</p> <p>simple CLI shell, S-E</p> <p>sq, uhl file compression programs, S-E</p> <p>Yachio a familiar game, S-E</p> <p>Make a simple 'make' programming utility, S-E</p> <p>Emacs an early version of the Amiga text editor, S-E-D</p> <p>Assembler programs:</p> <p>bssearch.asm binary search code</p> <p>qsor.asm Unix compatible qsort() function, source and C test program</p> <p>setmp.asm setmp() code for Lattice 3.02</p> <p>SVprint Unix system V compatible printer</p> <p>tees.c Unix compatible tree() function, O-D</p> <p>(This disk formerly had IFF specification files and examples. Since this spec is constantly updated, the IFF spec files have been moved to their own disk in the AMICUS collection.)</p> <p>John Draper Amiga Tutorial:</p> <p>Animate describes animation algorithms</p> <p>Gadgets tutorial on gadgets</p> <p>Menus learn about intuition menus</p>	<p>AMICUS Disk 3</p> <p>C programs:</p> <p>Xref a C cross-reference gen., S-E</p> <p>30color extra-half-bright chip gfx demo, S-E</p> <p>truncate (chop) files down to size, S-E</p> <p>CleanUp removes strange characters from text files</p> <p>CR2LP converts carriage returns to line feeds in Amiga files, S-E</p> <p>Error adds compile errors to a C file, S</p> <p>Helo window ex. from the RKM, S</p> <p>Kermit generic Kermit implementation, flakey, no terminal mode, S-E</p> <p>Scales sound demo plays scales, S-E</p> <p>SkewB Rubik cube demo in hi-res colors, S-E</p> <p>AmigaBasicProg(dir)</p> <p>Autmata cellular automata simulation</p> <p>CrayEight card game</p> <p>Graph function graphing programs</p> <p>WinningHour a game</p> <p>Basic programs:</p> <p>Casino games of poker, blackjack, dice, and craps</p> <p>Gomoku also known as 'othello'</p> <p>Sabotege sort of an adventure game</p> <p>Executable programs:</p> <p>Disassem a 68020 disassembler, E-D</p> <p>DpSlide shows a given set of IFF pictures, E-D</p> <p>Arrange a text formatting program, E-D</p> <p>Assembler programs:</p> <p>Argoterm terminal program with speech and Xmodem, S-E</p>	<p>AMICUS Disk 4</p> <p>Files from the original Amiga Technical BBS</p> <p>Note that some of these files are old, and refer to older versions of the operating system. These files came from the Sun system that served as Amiga technical support HQ for most of 1985. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.</p> <p>Complete and nearly up-to-date C source to 'Image.ed', an early version of the Icon Editor. This is a little flakey, but compiles and runs.</p> <p>An Intuition demo, in full C source, including files: demomenu.c, demomenu2.c, demoreq.c, getasol.c, idemo.c, idemo guide, idemo make, idemoall.h, idemo.c, and towrite.c</p> <p>addmem.c add external memory to the system</p> <p>example of BOB use</p> <p>consoleIO.c console IO example</p> <p>createport.c create and delete ports</p> <p>create standard IO requests</p> <p>creatask.c creating task examples</p> <p>example of track read and write</p> <p>source to the 'doty window' demo</p> <p>dualplay.c dual playfield example</p> <p>flood.c flood fill example</p> <p>freemap.c old version of 'teemap'</p> <p>tools for VSprites and BOBs</p> <p>graphic memory usage indicator</p> <p>window example from RKM</p> <p>adding an input handler to the input stream</p> <p>reading the joystick</p> <p>direct keyboard reading</p> <p>layers examples</p> <p>test mouse port</p> <p>ownlib.c</p> <p>ownlib.asm</p> <p>paratest.c</p>	<p>AMICUS Disk 5</p> <p>Files from the Amiga Link / Amiga Information Network</p> <p>Note that some of these files are old, and refer to older versions of the operating system. These files are from AmigaLink. For a time, Commodore supported AmigaLink, aka AIN, for online developer technical support. It was only up and running for several weeks. These files do not carry a warranty, and are for educational purposes only. Of course, that's not to say they don't work.</p> <p>A demo of Intuition menus called 'menudemo', in C source</p> <p>whereis.c find a file searching all subdirectories</p> <p>bobtest.c BOB programming example</p> <p>sewep.c sound synthesis example</p> <p>Assembler files:</p> <p>mydev.asm sample device driver</p> <p>mylib.asm sample library example</p> <p>mylib.c</p> <p>mydev.i</p> <p>asm-suppl</p> <p>macros.i assembler include files</p> <p>Tests:</p> <p>amigaticks tips on CLI commands</p> <p>extdisk external disk specification</p> <p>gameport game port spec</p> <p>parallel port parallel port spec</p> <p>serial serial port spec</p> <p>1.1.update list of new features in version 1.1</p> <p>1.1.txt 'diff' of include file changes from version 1.0 to 1.1</p> <p>Files for building your own printer drivers, including dospal.c, eposdata.c, initasm, printer.c, printer.link, printtag.asm, render.c, and waitasm. This disk does contain a number of files describing the IFF specification. These are not the latest and greatest files, but remain here for historical purposes. They include text files and C source examples. The latest IFF spec is elsewhere in this library.</p>	<p>AMICUS Disk 6</p> <p>IFF Pictures</p> <p>This disk includes the DfSide program, which can view a given series of IFF pictures, and the 'showpic' program, which can view each file by the click of an icon. The pictures include a screen from AricFox, a Degas dancer, the guys at Electronic Arts, a gorilla, horses, King Tut, a lighthouse, a screen from Marble Madness, the Bugs Bunny Marfan, a still from an old movie, the Dire Straits moving company, a screen from Pinball Construction Set, a TV newscaster, the PaintCan, a world map, a Porsche, a shuttle mission path, a tyrannosaurus rex, a planet view, a VISA card, and a ten-speed.</p> <p>AMICUS Disk 7</p> <p>DigView HAM demo picture disk</p> <p>This disk has pictures from the DigView hold-and-modify video digitizer. It includes the ladies with pencils and lollypops, the young girl, the bulldozer, the horse and buggy, the Byte cover, the dictionary page, the robot and Robert. This includes a program to view each picture separately, and all together as separate, slidable screens. The 'seelion' program, to turn any screen into an IFF picture.</p> <p>AMICUS Disk 8</p> <p>C programs:</p> <p>Browse view text files on a disk, using menus S-E-D</p> <p>Crunch removes comments and white space from C files, S-E</p> <p>IconExec EXECUTE a series of commands from Workbench S-E</p> <p>PDScreen Dumps Rastport of highest screen to printer</p> <p>SetAlternate sets a second image for an icon, when clicked once S-E</p> <p>SetWindow makes windows for a CLI program to run under Workbench S-E</p> <p>SmallClock a small digital clock in a window menu bar</p> <p>Scripper the screen printer in the fourth AC S-E</p>	<p>Amiga Basic Programs:</p> <p>(Note: Many of these programs are present on AMICUS Disk 1. Several of these were converted to Amiga Basic, and are included here.)</p> <p>AddressBook a simple address book database</p> <p>Ball draws a ball</p> <p>Cloud program to convert CompuServe hex files to binary, S-D</p> <p>Cue the game, Intuition driven</p> <p>ColorArt art drawing program</p> <p>DeluxeDraw the drawing program in the 3rd AC, S-D</p> <p>Eliza conversational computer psychologist</p> <p>Othello the game, as known as 'go'</p> <p>RatMaze 3D ratmaze game</p> <p>ROR bogging graphics demo</p> <p>Shuttle draws 3D pictures of the space shuttle</p> <p>Spelling simple spelling program</p> <p>YoYo word zero-gravity yo-yo demo, tracks yo-yo to the mouse</p> <p>Executable programs:</p> <p>3Dcube Modula-2 demo of a rotating cube</p> <p>AtIcon sets a second icon image, displayed when the icon is clicked</p> <p>AmigaSpell a slow but simple spell checker, E-D</p> <p>arc the ARC file compression program</p> <p>Bertrand graphics demo</p> <p>disksalvage prog. to rescue trashed disks, E-D</p> <p>KwikCopy a quick but nasty disk copy program</p> <p>LibDr lists hunks in an object file E-D</p> <p>SaveLibM saves any screen as IFF pic E-D ??</p> <p>ScreenDump shareware screen dump prog. Early version 2.0, term program, Xmodem-E-D</p> <p>StarTerm</p> <p>Tests:</p> <p>LetterMain tips on fixing .main.c in Lattice</p> <p>GDskDrive make your own 5 1/4 drive</p> <p>GuruMed explains the Guru numbers</p> <p>Let's30bugs bug list of Lattice C version 3.03</p> <p>MForgeRev user's view of the MicroForge HD</p> <p>PrintSpooler EXECUTE-based print spool prog.</p> <p>.BMAP files:</p> <p>These are the necessary links between Amiga Basic and the system libraries. To take advantage of the Amiga's capabilities in Basic, you need these files. BMAPs are included for 'disk', 'console', 'diskfont', 'exec', 'icon', 'intuition', 'layers', 'mathfp', 'mathvee', 'mathvec', 'mathvec2d', 'mathvec3d', 'mathvec4d', 'mathvec5d', 'mathvec6d', 'mathvec7d', 'mathvec8d', 'mathvec9d', 'mathvec10d', 'mathvec11d', 'mathvec12d', 'mathvec13d', 'mathvec14d', 'mathvec15d', 'mathvec16d', 'mathvec17d', 'mathvec18d', 'mathvec19d', 'mathvec20d', 'mathvec21d', 'mathvec22d', 'mathvec23d', 'mathvec24d', 'mathvec25d', 'mathvec26d', 'mathvec27d', 'mathvec28d', 'mathvec29d', 'mathvec30d', 'mathvec31d', 'mathvec32d', 'mathvec33d', 'mathvec34d', 'mathvec35d', 'mathvec36d', 'mathvec37d', 'mathvec38d', 'mathvec39d', 'mathvec40d', 'mathvec41d', 'mathvec42d', 'mathvec43d', 'mathvec44d', 'mathvec45d', 'mathvec46d', 'mathvec47d', 'mathvec48d', 'mathvec49d', 'mathvec50d', 'mathvec51d', 'mathvec52d', 'mathvec53d', 'mathvec54d', 'mathvec55d', 'mathvec56d', 'mathvec57d', 'mathvec58d', 'mathvec59d', 'mathvec60d', 'mathvec61d', 'mathvec62d', 'mathvec63d', 'mathvec64d', 'mathvec65d', 'mathvec66d', 'mathvec67d', 'mathvec68d', 'mathvec69d', 'mathvec70d', 'mathvec71d', 'mathvec72d', 'mathvec73d', 'mathvec74d', 'mathvec75d', 'mathvec76d', 'mathvec77d', 'mathvec78d', 'mathvec79d', 'mathvec80d', 'mathvec81d', 'mathvec82d', 'mathvec83d', 'mathvec84d', 'mathvec85d', 'mathvec86d', 'mathvec87d', 'mathvec88d', 'mathvec89d', 'mathvec90d', 'mathvec91d', 'mathvec92d', 'mathvec93d', 'mathvec94d', 'mathvec95d', 'mathvec96d', 'mathvec97d', 'mathvec98d', 'mathvec99d', 'mathvec100d', 'mathvec101d', 'mathvec102d', 'mathvec103d', 'mathvec104d', 'mathvec105d', 'mathvec106d', 'mathvec107d', 'mathvec108d', 'mathvec109d', 'mathvec110d', 'mathvec111d', 'mathvec112d', 'mathvec113d', 'mathvec114d', 'mathvec115d', 'mathvec116d', 'mathvec117d', 'mathvec118d', 'mathvec119d', 'mathvec120d', 'mathvec121d', 'mathvec122d', 'mathvec123d', 'mathvec124d', 'mathvec125d', 'mathvec126d', 'mathvec127d', 'mathvec128d', 'mathvec129d', 'mathvec130d', 'mathvec131d', 'mathvec132d', 'mathvec133d', 'mathvec134d', 'mathvec135d', 'mathvec136d', 'mathvec137d', 'mathvec138d', 'mathvec139d', 'mathvec140d', 'mathvec141d', 'mathvec142d', 'mathvec143d', 'mathvec144d', 'mathvec145d', 'mathvec146d', 'mathvec147d', 'mathvec148d', 'mathvec149d', 'mathvec150d', 'mathvec151d', 'mathvec152d', 'mathvec153d', 'mathvec154d', 'mathvec155d', 'mathvec156d', 'mathvec157d', 'mathvec158d', 'mathvec159d', 'mathvec160d', 'mathvec161d', 'mathvec162d', 'mathvec163d', 'mathvec164d', 'mathvec165d', 'mathvec166d', 'mathvec167d', 'mathvec168d', 'mathvec169d', 'mathvec170d', 'mathvec171d', 'mathvec172d', 'mathvec173d', 'mathvec174d', 'mathvec175d', 'mathvec176d', 'mathvec177d', 'mathvec178d', 'mathvec179d', 'mathvec180d', 'mathvec181d', 'mathvec182d', 'mathvec183d', 'mathvec184d', 'mathvec185d', 'mathvec186d', 'mathvec187d', 'mathvec188d', 'mathvec189d', 'mathvec190d', 'mathvec191d', 'mathvec192d', 'mathvec193d', 'mathvec194d', 'mathvec195d', 'mathvec196d', 'mathvec197d', 'mathvec198d', 'mathvec199d', 'mathvec200d', 'mathvec201d', 'mathvec202d', 'mathvec203d', 'mathvec204d', 'mathvec205d', 'mathvec206d', 'mathvec207d', 'mathvec208d', 'mathvec209d', 'mathvec210d', 'mathvec211d', 'mathvec212d', 'mathvec213d', 'mathvec214d', 'mathvec215d', 'mathvec216d', 'mathvec217d', 'mathvec218d', 'mathvec219d', 'mathvec220d', 'mathvec221d', 'mathvec222d', 'mathvec223d', 'mathvec224d', 'mathvec225d', 'mathvec226d', 'mathvec227d', 'mathvec228d', 'mathvec229d', 'mathvec230d', 'mathvec231d', 'mathvec232d', 'mathvec233d', 'mathvec234d', 'mathvec235d', 'mathvec236d', 'mathvec237d', 'mathvec238d', 'mathvec239d', 'mathvec240d', 'mathvec241d', 'mathvec242d', 'mathvec243d', 'mathvec244d', 'mathvec245d', 'mathvec246d', 'mathvec247d', 'mathvec248d', 'mathvec249d', 'mathvec250d', 'mathvec251d', 'mathvec252d', 'mathvec253d', 'mathvec254d', 'mathvec255d', 'mathvec256d', 'mathvec257d', 'mathvec258d', 'mathvec259d', 'mathvec260d', 'mathvec261d', 'mathvec262d', 'mathvec263d', 'mathvec264d', 'mathvec265d', 'mathvec266d', 'mathvec267d', 'mathvec268d', 'mathvec269d', 'mathvec270d', 'mathvec271d', 'mathvec272d', 'mathvec273d', 'mathvec274d', 'mathvec275d', 'mathvec276d', 'mathvec277d', 'mathvec278d', 'mathvec279d', 'mathvec280d', 'mathvec281d', 'mathvec282d', 'mathvec283d', 'mathvec284d', 'mathvec285d', 'mathvec286d', 'mathvec287d', 'mathvec288d', 'mathvec289d', 'mathvec290d', 'mathvec291d', 'mathvec292d', 'mathvec293d', 'mathvec294d', 'mathvec295d', 'mathvec296d', 'mathvec297d', 'mathvec298d', 'mathvec299d', 'mathvec300d', 'mathvec301d', 'mathvec302d', 'mathvec303d', 'mathvec304d', 'mathvec305d', 'mathvec306d', 'mathvec307d', 'mathvec308d', 'mathvec309d', 'mathvec310d', 'mathvec311d', 'mathvec312d', 'mathvec313d', 'mathvec314d', 'mathvec315d', 'mathvec316d', 'mathvec317d', 'mathvec318d', 'mathvec319d', 'mathvec320d', 'mathvec321d', 'mathvec322d', 'mathvec323d', 'mathvec324d', 'mathvec325d', 'mathvec326d', 'mathvec327d', 'mathvec328d', 'mathvec329d', 'mathvec330d', 'mathvec331d', 'mathvec332d', 'mathvec333d', 'mathvec334d', 'mathvec335d', 'mathvec336d', 'mathvec337d', 'mathvec338d', 'mathvec339d', 'mathvec340d', 'mathvec341d', 'mathvec342d', 'mathvec343d', 'mathvec344d', 'mathvec345d', 'mathvec346d', 'mathvec347d', 'mathvec348d', 'mathvec349d', 'mathvec350d', 'mathvec351d', 'mathvec352d', 'mathvec353d', 'mathvec354d', 'mathvec355d', 'mathvec356d', 'mathvec357d', 'mathvec358d', 'mathvec359d', 'mathvec360d', 'mathvec361d', 'mathvec362d', 'mathvec363d', 'mathvec364d', 'mathvec365d', 'mathvec366d', 'mathvec367d', 'mathvec368d', 'mathvec369d', 'mathvec370d', 'mathvec371d', 'mathvec372d', 'mathvec373d', 'mathvec374d', 'mathvec375d', 'mathvec376d', 'mathvec377d', 'mathvec378d', 'mathvec379d', 'mathvec380d', 'mathvec381d', 'mathvec382d', 'mathvec383d', 'mathvec384d', 'mathvec385d', 'mathvec386d', 'mathvec387d', 'mathvec388d', 'mathvec389d', 'mathvec390d', 'mathvec391d', 'mathvec392d', 'mathvec393d', 'mathvec394d', 'mathvec395d', 'mathvec396d', 'mathvec397d', 'mathvec398d', 'mathvec399d', 'mathvec400d', 'mathvec401d', 'mathvec402d', 'mathvec403d', 'mathvec404d', 'mathvec405d', 'mathvec406d', 'mathvec407d', 'mathvec408d', 'mathvec409d', 'mathvec410d', 'mathvec411d', 'mathvec412d', 'mathvec413d', 'mathvec414d', 'mathvec415d', 'mathvec416d', 'mathvec417d', 'mathvec418d', 'mathvec419d', 'mathvec420d', 'mathvec421d', 'mathvec422d', 'mathvec423d', 'mathvec424d', 'mathvec425d', 'mathvec426d', 'mathvec427d', 'mathvec428d', 'mathvec429d', 'mathvec430d', 'mathvec431d', 'mathvec432d', 'mathvec433d', 'mathvec434d', 'mathvec435d', 'mathvec436d', 'mathvec437d', 'mathvec438d', 'mathvec439d', 'mathvec440d', 'mathvec441d', 'mathvec442d', 'mathvec443d', 'mathvec444d', 'mathvec445d', 'mathvec446d', 'mathvec447d', 'mathvec448d', 'mathvec449d', 'mathvec450d', 'mathvec451d', 'mathvec452d', 'mathvec453d', 'mathvec454d', 'mathvec455d', 'mathvec456d', 'mathvec457d', 'mathvec458d', 'mathvec459d', 'mathvec460d', 'mathvec461d', 'mathvec462d', 'mathvec463d', 'mathvec464d', 'mathvec465d', 'mathvec466d', 'mathvec467d', 'mathvec468d', 'mathvec469d', 'mathvec470d', 'mathvec471d', 'mathvec472d', 'mathvec473d', 'mathvec474d', 'mathvec475d', 'mathvec476d', 'mathvec477d', 'mathvec478d', 'mathvec479d', 'mathvec480d', 'mathvec481d', 'mathvec482d', 'mathvec483d', 'mathvec484d', 'mathvec485d', 'mathvec486d', 'mathvec487d', 'mathvec488d', 'mathvec489d', 'mathvec490d', 'mathvec491d', 'mathvec492d', 'mathvec493d', 'mathvec494d', 'mathvec495d', 'mathvec496d', 'mathvec497d', 'mathvec498d', 'mathvec499d', 'mathvec500d', 'mathvec501d', 'mathvec502d', 'mathvec503d', 'mathvec504d', 'mathvec505d', 'mathvec506d', 'mathvec507d', 'mathvec508d', 'mathvec509d', 'mathvec510d', 'mathvec511d', 'mathvec512d', 'mathvec513d', 'mathvec514d', 'mathvec515d', 'mathvec516d', 'mathvec517d', 'mathvec518d', 'mathvec519d', 'mathvec520d', 'mathvec521d', 'mathvec522d', 'mathvec523d', 'mathvec524d', 'mathvec525d', 'mathvec526d', 'mathvec527d', 'mathvec528d', 'mathvec529d', 'mathvec530d', 'mathvec531d', 'mathvec532d', 'mathvec533d', 'mathvec534d', 'mathvec535d', 'mathvec536d', 'mathvec537d', 'mathvec538d', 'mathvec539d', 'mathvec540d', 'mathvec541d', 'mathvec542d', 'mathvec543d', 'mathvec544d', 'mathvec545d', 'mathvec546d', 'mathvec547d', 'mathvec548d', 'mathvec549d', 'mathvec550d', 'mathvec551d', 'mathvec552d', 'mathvec553d', 'mathvec554d', 'mathvec555d', 'mathvec556d', 'mathvec557d', 'mathvec558d', 'mathvec559d', 'mathvec560d', 'mathvec561d', 'mathvec562d', 'mathvec563d', 'mathvec564d', 'mathvec565d', 'mathvec566d', 'mathvec567d', 'mathvec568d', 'mathvec569d', 'mathvec570d', 'mathvec571d', 'mathvec572d', 'mathvec573d', 'mathvec574d', 'mathvec575d', 'mathvec576d', 'mathvec577d', 'mathvec578d', 'mathvec579d', 'mathvec580d', 'mathvec581d', 'mathvec582d', 'mathvec583d', 'mathvec584d', 'mathvec585d', 'mathvec586d', 'mathvec587d', 'mathvec588d', 'mathvec589d', 'mathvec590d', 'mathvec591d', 'mathvec592d', 'mathvec593d', 'mathvec594d', 'mathvec595d', 'mathvec596d', 'mathvec597d', 'mathvec598d', 'mathvec599d', 'mathvec600d', 'mathvec601d', 'mathvec602d', 'mathvec603d', 'mathvec604d', 'mathvec605d', 'mathvec606d', 'mathvec607d', 'mathvec608d', 'mathvec609d', 'mathvec610d', 'mathvec611d', 'mathvec612d', 'mathvec613d', 'mathvec614d', 'mathvec615d', 'mathvec616d', 'mathvec617d', 'mathvec618d', 'mathvec619d', 'mathvec620d', 'mathvec621d', 'mathvec622d', 'mathvec623d', 'mathvec624d', 'mathvec625d', 'mathvec626d', 'mathvec627d', 'mathvec628d', 'mathvec629d', 'mathvec630d', 'mathvec631d', 'mathvec632d', 'mathvec633d', 'mathvec634d', 'mathvec635d', 'mathvec636d', 'mathvec637d', 'mathvec638d', 'mathvec639d', 'mathvec640d', 'mathvec641d', 'mathvec642d', 'mathvec643d', 'mathvec644d', 'mathvec645d', 'mathvec646d', 'mathvec647d', 'mathvec648d', 'mathvec649d', 'mathvec650d', 'mathvec651d', 'mathvec652d', 'mathvec653d', 'mathvec654d', 'mathvec655d', 'mathvec656d', 'mathvec657d', 'mathvec658d', 'mathvec659d', 'mathvec660d', 'mathvec661d', 'mathvec662d', 'mathvec663d', 'mathvec664d', 'mathvec665d', 'mathvec666d', 'mathvec667d', 'mathvec668d', 'mathvec669d', 'mathvec670d', 'mathvec671d', 'mathvec672d', 'mathvec673d', 'mathvec674d', 'mathvec675d', 'mathvec676d', 'mathvec677d', 'mathvec678d', 'mathvec679d', 'mathvec680d', 'mathvec681d', 'mathvec682d', 'mathvec683d', 'mathvec684d', 'mathvec685d', 'mathvec686d', 'mathvec687d', 'mathvec688d', 'mathvec689d', 'mathvec690d', 'mathvec691d', 'mathvec692d', 'mathvec693d', 'mathvec694d', 'mathvec695d', 'mathvec696d', 'mathvec697d', 'mathvec698d', 'mathvec699d', 'mathvec700d', 'mathvec701d', 'mathvec702d', 'mathvec703d', 'mathvec704d', 'mathvec705d', 'mathvec706d', 'mathvec707d', 'mathvec708d', 'mathvec709d', 'mathvec710d', 'mathvec711d', 'mathvec712d', 'mathvec713d', 'mathvec714d', 'mathvec715d', 'mathvec716d', 'mathvec717d', 'mathvec718d', 'mathvec719d', 'mathvec720d', 'mathvec721d', 'mathvec722d', 'mathvec723d', 'mathvec724d', 'mathvec725d', 'mathvec726d', 'mathvec727d', 'mathvec728d', 'mathvec729d', 'mathvec730d', 'mathvec731d', 'mathvec732d', 'mathvec733d', 'mathvec734d', 'mathvec735d', 'mathvec736d', 'mathvec737d', 'mathvec738d', 'mathvec739d', 'mathvec740d', 'mathvec741d', 'mathvec742d', 'mathvec743d', 'mathvec744d', 'mathvec745d', 'mathvec746d', 'mathvec747d', 'mathvec748d', 'mathvec749d', 'mathvec750d', 'mathvec751d', 'mathvec752d', 'mathvec753d', 'mathvec754d', 'mathvec755d', 'mathvec756d', 'mathvec757d', 'mathvec758d', 'mathvec759d', 'mathvec760d', 'mathvec761d', 'mathvec762d', 'mathvec763d', 'mathvec764d', 'mathvec765d', 'mathvec766d', 'mathvec767d', 'mathvec768d', 'mathvec769d', 'mathvec770d', 'mathvec771d', 'mathvec772d', 'mathvec773d', 'mathvec774d', 'mathvec775d', 'mathvec776d', 'mathvec777d', 'mathvec778d', 'mathvec779d', 'mathvec780d', 'mathvec781d', 'mathvec782d', 'mathvec783d', 'mathvec784d', 'mathvec785d', 'mathvec786d', 'mathvec787d', 'mathvec788d', 'mathvec789d', 'mathvec790d', 'mathvec791d', 'mathvec792d', 'mathvec793d', 'mathvec794d', 'mathvec795d', 'mathvec796d', 'mathvec797d', 'mathvec798d', 'mathvec799d', 'mathvec800d', 'mathvec801d', 'mathvec802d', 'mathvec803d', 'mathvec804d', 'mathvec805d', 'mathvec806d', 'mathvec807d', 'mathvec808d', 'mathvec809d', 'mathvec810d', 'mathvec811d', 'mathvec812d', 'mathvec813d', 'mathvec814d', 'mathvec815d', 'mathvec816d', 'mathvec817d', 'mathvec818d', 'mathvec819d', 'mathvec820d', 'mathvec821d', 'mathvec822d', 'mathvec823d', 'mathvec824d', 'mathvec825d', 'mathvec826d', 'mathvec827d', 'mathvec828d', 'mathvec829d', 'mathvec830d', 'mathvec831d', 'mathvec832d', 'mathvec833d', 'mathvec834d', 'mathvec835d', 'mathvec836d', 'mathvec837d', 'mathvec838d', 'mathvec839d', 'mathvec840d', 'mathvec841d', 'mathvec842d', 'mathvec843d', 'mathvec844d', 'mathvec845d', 'mathvec846d', 'mathvec847d', 'mathvec848d', 'mathvec849d', 'mathvec850d', 'mathvec851d', 'mathvec852d', 'mathvec853d', 'mathvec854d', 'mathvec855d', 'mathvec856d', 'mathvec857d', 'mathvec858d', 'mathvec859d', 'mathvec860d', 'mathvec861d', 'mathvec862d', 'mathvec863d', 'mathvec864d', 'mathvec865d', 'mathvec866d', 'mathvec867d', 'mathvec868d', 'mathvec869d', 'mathvec870d', 'mathvec871d', 'mathvec872d', 'mathvec873d', 'mathvec874d', 'mathvec875d', 'mathvec876d', 'mathvec877d', 'mathvec878d', 'mathvec879d', 'mathvec880d', 'mathvec881d', 'mathvec882d', 'mathvec883d', 'mathvec884d', 'mathvec885d', 'mathvec886d', 'mathvec887d', 'mathvec888d', 'mathvec889d', 'mathvec890d', 'mathvec891d', 'mathvec892d', 'mathvec893d', 'mathvec894d', 'mathvec895d', 'mathvec896d', 'mathvec897d', 'mathvec898d', 'mathvec899d', 'mathvec900d', 'mathvec901d', 'mathvec902d', 'mathvec903d', 'mathvec904d', 'mathvec905d', 'mathvec906d', 'mathvec907d', 'mathvec908d', 'mathvec909d', 'mathvec910d', 'mathvec911d', 'mathvec912d', 'mathvec913d', 'mathvec914d', 'mathvec915d', 'mathvec916d', 'mathvec917d', 'mathvec918d', 'mathvec919d', 'mathvec920d', 'mathvec921d', 'mathvec922d', 'mathvec923d', 'mathvec924d', 'mathvec925d', 'mathvec926d', 'mathvec927d', 'mathvec928d', 'mathvec929d', 'mathvec930d', 'mathvec931d', 'mathvec932d', 'mathvec933d', 'mathvec934d', 'mathvec935d', 'mathvec936d', 'mathvec937d', 'mathvec938d', 'mathvec939d', 'mathvec940d', 'mathvec941d', 'mathvec942d', 'mathvec943d', 'mathvec944d', 'mathvec945d', 'mathvec946d', 'mathvec947d', 'mathvec948d', 'mathvec949d', 'mathvec950d', 'mathvec951d', 'mathvec952d', 'mathvec953d', 'mathvec954d', 'mathvec955d', 'mathvec956d', 'mathvec957d', 'mathvec958d', 'mathvec959d', 'mathvec960d', 'mathvec961d', 'mathvec962d', 'mathvec963d', 'mathvec964d', 'mathvec965d', 'mathvec966d', 'mathvec967d', 'mathvec968d', 'mathvec969d', 'mathvec970d', 'mathvec971d', 'mathvec972d', 'mathvec973d', 'mathvec974d', 'mathvec975d', 'mathvec976d', 'mathvec977d', 'mathvec978d', 'mathvec979d', 'mathvec980d', 'mathvec981d', 'mathvec982d', 'mathvec983d', 'mathvec984d', 'mathvec985d', 'mathvec986d', 'mathvec987d', 'mathvec988d', 'mathvec989d', 'mathvec990d', 'mathvec991d', 'mathvec992d', 'mathvec993d', 'mathvec994d', 'mathvec995d', 'mathvec996d', 'mathvec997d', 'mathvec998d', 'mathvec999d', 'mathvec1000d', 'mathvec1001d', 'mathvec1002d', 'mathvec1003d', 'mathvec1004d', 'mathvec1005d', 'mathvec1006d', 'mathvec1007d', 'mathvec1008d', 'mathvec1009d', 'mathvec1010d', 'mathvec1011d', 'mathvec1012d', 'mathvec1013d', 'mathvec1014d', 'mathvec1015d', 'mathvec1016d', 'mathvec1017d', 'mathvec1018d', 'mathvec1019d', 'mathvec1020d', 'mathvec1021d', 'mathvec1022d', 'mathvec1023d', 'mathvec1024d', 'mathvec1025d', 'mathvec1026d', 'mathvec1027d', 'mathvec1028d', 'mathvec1029d', 'mathvec1030d', 'mathvec1031d', 'mathvec1032d', 'mathvec1033d', 'mathvec1034d', 'mathvec1035d', 'mathvec1036d', 'mathvec1037d', 'mathvec1038d', 'mathvec1039d', 'mathvec1040d', 'mathvec1041d', 'mathvec1042d', 'mathvec1043d', 'mathvec1044d', 'mathvec1045d', 'mathvec1046d', 'mathvec1047d', 'mathvec1048d', 'mathvec1049d', 'mathvec1050d', 'mathvec1051d', 'mathvec1052d', 'mathvec1053d', 'mathvec1054d', 'mathvec1055d', 'mathvec1056d', 'mathvec1057d', 'mathvec1058d', 'mathvec1059d', 'mathvec1060d', 'mathvec1061d', 'mathvec1062d', 'mathvec1063d', 'mathvec1064d', 'mathvec1065d', 'mathvec1066d', 'mathvec1067d', 'mathvec1068d', 'mathvec1069d', 'mathvec1070d', 'mathvec1071d', 'mathvec1072d', 'mathvec1073d', 'mathvec1074d', 'mathvec1075d', 'mathvec1076d', 'mathvec1077d', 'mathvec1078d', 'mathvec1079d', 'mathvec1080d', 'mathvec1081d', 'mathvec1082d', 'mathvec1083d', 'mathvec1084d', 'mathvec1085d', 'mathvec1086d', 'mathvec1087d', 'mathvec1088d', 'mathvec1089d', 'mathvec1090d', 'mathvec1091d', 'mathvec1092d', 'mathvec1093d', 'mathvec1094d', 'mathvec1095d', 'mathvec1096d', 'mathvec1097d', 'mathvec1098d', 'mathvec1099d',</p>
---	--	---	--	---	--	--

<p>Texts:</p> <p>FrncKeys explains how to read function keys from Amiga Basic</p> <p>HeckerSin explains how to win the game 'hacker' guide to installing a 68010 in your Amiga</p> <p>PrinterTip sending escape sequences to your printer tips on setting up your startup-sequence file list of Transformer programs that work</p> <p>Printer Drivers: Printer drivers for the Canon PJ-1080A, the C-Itch Prowriter, an improved Epson driver that eliminates streaking, the Epson LQ-800, the Gemini Star-10, the NEC 8025A, the Okidata ML-92, the Panasonic KX-P100x family, and the Smith-Corona D300, with a document describing the installation process.</p> <p>AMICUS Disk 10 Instrument sound demo</p> <p>This is an icon-driven demo, circulated to many dealers. It includes the sounds of an acoustic guitar, an alarm, a banjo, a bass guitar, a bink, a callopie, a car horn, claves, water drip, electric guitar, a flute, a harp arpeggio, a kiddrum, a marimba, a organ minor chord, people talking, pigs, a pipe organ, a Rhodes piano, a saxophone, a sitar, a snare drum, a steel drum, bells, a vibraphone, a violin, a wailing guitar, a horse whinny, and a whistle.</p> <p>AMICUS Disk 11 C programs</p> <p>cpri Intuition-based, CLI replacement manager</p> <p>ps shows and adjusts priority of CLI processes, S-E</p> <p>ps shows info on CLI processes, S-E</p> <p>ps displays Compuserve RLE pics, S-E</p> <p>AmigaBasic programs</p> <p>pointed pointer and sprite editor program</p> <p>optimize optimization example from AC article</p> <p>caendar large, animated calendar, diary and date book program</p> <p>amortize loan amortizations</p> <p>brush2BOB converts small IFF brushes to AmigaBasic BOB OBJECTS</p> <p>grids draw and play waveforms</p> <p>hilbert draws Hilbert curves</p> <p>medib mad lib story generator</p> <p>mailtalk talking mailing list program</p> <p>meadows3D 3D graphics program, from a C™ article</p> <p>mcusetack mouse tracking example in hires mode</p> <p>slot slot machine game</p> <p>scacoe the game</p> <p>switch pachinko-like game</p> <p>weird makes strange sounds</p> <p>Executable programs</p> <p>cp unix-like copy command, E</p> <p>cls screen clear, S-E</p> <p>diff unix-like stream editor uses 'diff' output to fix files</p> <p>pm chart recorder performances indicator</p> <p>Assembler programs</p> <p>cls screen clear and CLI arguments example</p> <p>Modula-2</p> <p>trails moving-worm graphics demo</p> <p>caseconvert converts Modula-2 keywords to uppercase</p> <p>Forth Bresenham circle algorithm example</p> <p>Analyze 12 templates for the spreadsheet. Analyze</p> <p>There are four programs here that read Commodore 64 picture files. They can translate Koala Pad, Doodle, Print Shop and News Room graphics to IFF format. Getting the files from your C-64 to your Amiga is the hard part.</p> <p>AMICUS Disk 12</p> <p>Executable programs</p> <p>blink 'alink' compatible linker, but faster, E-D</p> <p>clean spins the disk for disk cleaners, E-D</p> <p>epsonset sends Epson settings to PAR from menu E-D</p> <p>showbig view hi-res pics in low-res superbmap, E-D</p> <p>speaktme tell the time, E-D</p> <p>undelete undeletes a file, E-D</p> <p>crvaphdm converts Apple][low, medium and high res pictures to IFF, E-D</p> <p>manued menu editor produces C code for menu, E-D</p> <p>quick quick disk-to-disk nibble copier, E-D</p> <p>quickIEA copies Electronic Arts disks, removes protection, E-D</p> <p>text 1.3 demo of text editor from Microsmiths, E-D</p> <p>C programs</p> <p>spin3 rotating blocks graphics demo, S-E-D</p> <p>ppopi start a new CLI at the press of a button, like Sidekick, S-E-D</p> <p>vsprite VSprite example code from Commodore, S-E-D</p> <p>AmigaBBS Amiga Basic bulletin board prog., S-D</p> <p>Assembler programs</p> <p>star10 makes star fields like Star Trek intro, S-E-D</p> <p>Pictures</p> <p>MountMandelbrot 3D view of Mandelbrot set</p> <p>Sar Destroyer hi-res Star Wars starship</p> <p>Robot arm robot arm grabbing a cylinder</p> <p>Texts</p> <p>windows Amiga vendors, names, addresses</p> <p>cardco fixes to early Cardco memory boards</p> <p>include cross-reference to C include files</p> <p>mindwalker clues to playing the game well</p> <p>sideshow make your own slideshows from the Kaleidoscope disk</p> <p>AMICUS Disk 13 Amiga Basic programs</p> <p>Routines from Carolyn Scheppner of CBM Tech Support, to read and display IFF pictures from Amiga Basic. With documentation. Also included is a program to do screen prints in Amiga Basic, and the newest BMAP files, with a corrected ConvertD program. With example pictures, and the SaveLBM screen capture program.</p> <p>Routines to load and play FutureSound and IFF sound files from Amiga Basic, by John Foust for Applied Visions. With</p>	<p>documentation and C and assembler source for writing your own libraries, and interfacing C to assembler in libraries. With example sound.</p> <p>Executable programs</p> <p>gravity Sci Amer Jan 86 gravitation graphic simulation, S-E-D</p> <p>Texts</p> <p>MIDI make your own MIDI instrument interface, with documentation and a hires schematic picture.</p> <p>AMICUS Disk 14 Several programs from Amazing Computing issues:</p> <p>Tools</p> <p>Dan Kary's C structure index program, S-E-D</p> <p>Amiga Basic programs:</p> <p>BMAP Reader by Tim Jones</p> <p>IFFBrush2BOB by Mike Swinger</p> <p>AutoRequester example</p> <p>DOSHelper Windowed help system for CLI commands, S-E-D</p> <p>PETrans translates PET ASCII files to ASCII files, S-E-D</p> <p>C Squared Graphics program from Scientific American, Sept 86, S-E-D</p> <p>orif adds or removes carriage returns from files, S-E-D</p> <p>dpdecode decrypts Deluxe Paint, remo</p> <p>vers copy protection, E-D</p> <p>queryWB asks Yes or No from the user, returns exit code, S-E</p> <p>vc VisiCalc type spreadsheet, no mouse control, E-D</p> <p>view views text files with window and gadget, E-D</p> <p>Olrig Olrig, Spooling, ysBoing, Zoing are sprite-based</p> <p>Boing! style demos, S-E-D</p> <p>CLIClock CLIClock, sClock, wClock are window border clocks, S-E-D</p> <p>Texts</p> <p>An article on long-persistence phosphor monitors, tips on making brushes of odd shapes in Deluxe Paint, and recommendations on icon interfaces from Commodore-Amiga.</p> <p>AMICUS Disk 15 The C programs include:</p> <p>'br' a file printing utility, which can print files in the background, and with line numbers and control character filtering.</p> <p>'tm' displays a chart of the blocks allocated on a disk</p> <p>'Ask' questions an 'execute' file, returns an error code to control the execution in that batch file</p> <p>'Stat' an enhanced version of AmigaDOS 'status' command.</p> <p>'Dissolve' random-dot dissolve demo displays IFF picture slowly, dot by dot, in a random fashion.</p> <p>PopCLIZ invoke new CLI window at the press of a key.</p> <p>The executable programs include:</p> <p>'Form' file formatting program through the printer driver to select print styles</p> <p>'DiskCat' catalogs disks, maintains, sorts, merges lists of disk files</p> <p>'PSound' SunRize Industries' sampled sound editor & recorder</p> <p>'Iconmaker' makes icons for most programs</p> <p>'Fractals' draws great fractal seascapes and mountain</p> <p>'3D Breakout' 3D glasses, create breakout in a new dimension</p> <p>'AmigaMonitor' displays lists of open files, tasks, devices and ports in use.</p> <p>'Commodors' version of 'asteroids' for the Amiga.</p> <p>'Sizzlers' high resolution graphics demo written in Modula 2.</p> <p>Texts:</p> <p>'ansi.txt' explains escape sequences the CON: device responds to.</p> <p>'FKey' includes template for making paper to sit in the tray at the top of the Amiga keyboard.</p> <p>'Spawn' programmer's document from Commodore</p> <p>Amiga, describe ways to use the Amiga's multitasking capabilities in your own programs.</p> <p>AmigaBasic programs:</p> <p>'Grids' draw sound waveforms, and hear them played.</p> <p>'Light' a version of the Tron light-cycle video game.</p> <p>'MgaSol' a game of solitaire.</p> <p>'Stats' program to calculate batting averages</p> <p>'Money' "try to grab all the bags of money that you can."</p> <p>AMICUS 15 also includes two beautiful IFF pictures, of the enemy walkers from the ice planet in Star Wars, and a picture of a cheetah.</p> <p>AMICUS Disk 16 'Juggler' demo by Eric Graham, a robot juggler bouncing three mirrored balls, with sound effects. Twenty-four frames of HAM animation are flipped quickly to produce this image. You control the speed of the juggling. The author's documentation hints that this program might someday be available as a product.</p> <p>IFF pictures</p> <p>parodies of the covers of Amiga World and Amazing Computing magazines.</p> <p>C programs:</p> <p>'InputHandler' example of making an input handler.</p> <p>'FileZip3' binary file editing program</p> <p>'ShowPrint' displays IFF picture, and prints it</p> <p>'Gen' program indexes and retrieves C structures and variables declared in the Amiga include file system.</p> <p>Executable Programs:</p> <p>'FixHunk2' repairs an executable program file for expanded memory</p> <p>'ms2smus' converts Music Studio files to IFF standard 'SMUS' format. I have heard this program might have a few bugs, especially in regards to very long songs, but it works in most cases.</p> <p>Amiga version of the 'Missile Command' video game.</p>	<p>This disk also contains several files of scenarios for Amiga Flight Simulator II. By putting one of these seven files on a blank disk, and inserting it in the drive after performing a special command in this game, a number of interesting locations are preset into the Flight Simulator program. For example, one scenario places your plane on Alcatraz, while another puts you in Central Park</p> <p>AMICUS Disk 17 Telecommunications disk which contains six terminal programs.</p> <p>'Comm' V1.33 term prog. with Xmodem, Wxmodem,</p> <p>'ATerm' V7.2 term prog. includes Super Kermit</p> <p>'VT-100' V2.6 Dave Wecker's VT-100 emulator with Xmodem, Kermit, and scripting</p> <p>'Amiga Kermit' V4D(060) port of the Unix C-Kermit</p> <p>'Vtek' V2.3.1 Tektronix graphics terminal emulator based on the VT-100 prog. V2.3 and contains latest 'arc' file compression</p> <p>'AmigaHost' V0.9 for Compuserve. Includes RLE graphics abilities & CIS-B file transfer protocol.</p> <p>'FixHunk' expansion memory necessity</p> <p>'FixObj' removes garbage characters from modern received files</p> <p>'Txt' filters text files from other systems to be read by the Amiga E.C.</p> <p>'addmem' executable version for use with mem expansion article in AC V2.1</p> <p>'arc' file documentation and a basic tutorial on un 'arc'ing files for making 'arc' files E.C.</p> <p>'arc' file documentation and a basic tutorial on un 'arc'ing files for making 'arc' files E.C.</p> <p>AMICUS Disk 18 Logo</p> <p>Amiga version of the popular computer language, with example programs, E-D</p> <p>TVText Demo version of the TVText character generator</p> <p>PageSetter Freely distributable versions of the updated PagePrint and PageIFF programs for the PageSetter desktop publishing package.</p> <p>FullWindow Resizes any CLI window using only CLI commands, E-D</p> <p>Life3d 3-D version of Conway's LIFE program, E-D</p> <p>Defdisk CLI utility to re-assign a new Workbench disk, S-E-D</p> <p>Calendar.WKS Lotus-compatible worksheet that makes calendars</p> <p>SetKey Demo of keyboard key re-programmer, with IFF picture to make function key labels, E-D</p> <p>VPG Video pattern generator for aligning monitors, E-D</p> <p>HP-10C Hewlett-Packard-like calculator, E-D</p> <p>SetPrefs Change the Preferences settings on the fly, in C, S-E-D</p> <p>StarProbe Program studies stellar evolution. C source included for Amiga and MS-DOS, S-E-D</p> <p>ROT C version of Colin French's AmigaBasic ROT program from Amazing Computing. ROT edits and displays polygons to create three dimensional objects. Up to 24 frames of animation can be created and displayed. E-D</p> <p>Scat Like Ing. windows on screen run away from the mouse, E-D</p> <p>DK Decays" the CLI window into dust, in Modula 2, S-E-D</p> <p>DropShadow2 Adds layered shadows to Workbench windows, E-D</p> <p>AMICUS Disk 19 This disk carries several programs from Amazing Computing. The IFF pictures on this disk include the Amiga Wake Part T-shirt logo, a sixteen-color hires image of Andy Griffith, and five Amiga Live! pictures from the Amazing Stories episode that featured the Amiga.</p> <p>Solve Linear equation solver in assembly language, S-E-D</p> <p>Gadgets Bryan Catley's AmigaBasicEditorial, Bryan Catley's AmigaBasic household inventory program, S-D</p> <p>Waveform Jim Shields' Waveform WofAmigaBasic, S-D</p> <p>DiskLib John Kennan's AmigaBasic disk librarian program, S-D</p> <p>Subscripts Ivan Smith's AmigaBasic subscript example, S-D</p> <p>String, Boolean C programs and executables for Harriet Maybeck Tolly's Intuition tutorials, S-E-D</p> <p>Skinny C Bob Remersma's example for making small C programs, S-E-D</p> <p>COMALh Make C look like COMAL H&D file, Makes Emacs function key definitions by Greg Douglas, S-D</p> <p>AMon 1.1 Snoop on system resource use, E-D</p> <p>Size Band's Tale character editor, E-D</p> <p>WnSize CLI program shows the size of a given set of files, E-D</p> <p>AMICUS Disk 20 Compactor, Decoder Steve Michel AmigaBasic tools, S-D</p> <p>BobEd BOB and sprite editor written in C, S-E-D</p> <p>SpriteMasterII Sprite editor and animator by Brad Kiefer, E-D</p> <p>BitLab Bitler chip exploration C program by Tomas Rokicki, S-E-D</p> <p>FPic Image processing program by Bob Bush loads and saves IFF images, changes them with several techniques, E-D</p> <p>Bankin Complete home banking program, balance your checkbook! E-D</p> <p>cons Console device demo program with supporting macro routines.</p> <p>treemap Creates a visual diagram of free memory</p> <p>inputdev sample input handler, traps key or mouse events</p>	<p>joystick Shows how to set up the gameport device as a joystick.</p> <p>keyboard demonstrates direct communications with the keyboard.</p> <p>layers Shows use of the layers library</p> <p>mandelbrot FF Mandelbrot program</p> <p>mouse hooks up mouse to right joystick port</p> <p>one window console window demo</p> <p>parallel Demonstrates access to the parallel port, opening and using the printer, does a screen dump, not working</p> <p>print support Printer support routines, not working</p> <p>protest sample process creation code, not working</p> <p>region demos split drawing regions</p> <p>samplefont sample font with info on creating your own</p> <p>serial Demos the serial port</p> <p>singlePlayfield Creates 320 x 200 playfield</p> <p>speechtoy latest version of cute speech demo</p> <p>speechdemo simplified version of speechtoy, with ID requests</p> <p>textdemo displays available fonts</p> <p>timer demos timer device use</p> <p>trackdisk demos trackdisk driver</p> <p>AMICUS Disk 21 Target</p> <p>Makes each mouse click sound like a gunshot, S-E-D</p> <p>Send Simple game of sand that follows the mouse pointer, E-D</p> <p>PropGadget Harriet Maybeck Tolly's proportional gadget example, S-E</p> <p>EHB Checks to see if you have extra-half-bright graphics, S-E-D</p> <p>Piano Simple piano sound program</p> <p>SciScripts Makes sci animation scripts for Aegis Animator, in AmigaBasic</p> <p>This disk has electronic catalogs for AMICUS disks 1 to 20 and Fun disks 1 to 60. They are viewed with the DiskCat program, included here.</p> <p>AMICUS Disk 22 Cycles</p> <p>Light cycle game, E-D</p> <p>Show_PrintII Views and prints IFF pictures, including larger than screen</p> <p>PrnDrvGen2.3 Video of a printer driver generator</p> <p>Animations VideoScape animations of planes and boing ball</p> <p>Garden Makes fractal gardenscapes</p> <p>BasicSorts Examples of binary search and insertion sort in AmigaBasic</p> <p>AMICUS Disk 23 An AMICUS disk completely dedicated to music on the Amiga. This disk contains two music players, songs, instruments, and players to bring the thrill of playing 'Big Sound' on your Amiga</p> <p>Instruments a collection of 26 instruments for playing and creating music. The collection ranges from Cannon to Marimba</p> <p>List INSTR program to list the instruments DMS will not load as well as list the origins for any instrument.</p> <p>Music a collection of 14 Classical pieces</p> <p>1812Overture The 16 minute classical feature complete with Cannon!</p> <p>Three Amiga Music Players</p> <p>SMUSPlay SMUSPlay</p> <p>MusicCra2SMUS MusicCra2SMUS</p> <p>MusicStudio2SMUS MusicStudio2SMUS</p> <p>AMICUS Disk 24 Sectors</p> <p>A disk sector editor for any AmigaDOS file-structured device, recover files from a trashed hard disk. By David Janer of MicroIllusions</p> <p>Iconize Reduces the size of IFF images, companion program, Recolor, remaps the palette colors of one picture to use the palette colors of another. Using these programs and a tool to convert IFF brushes to Workbench icons, make icons look like miniatures of the pictures.</p> <p>CodeDemo Modula-2 program converts assembler object files to inline CODE statements. Comes with a screen scrolling example</p> <p>AmiBug Workbench hack makes the same fly walk across the screen at random intervals. Otherwise, completely harmless.</p> <p>BNTTools Three examples of assembly language code from Bryce Nesbitt:</p> <p>1. SetLace prog to switch interface on/off.</p> <p>2. Why, replace AmigaDOS CLI Why</p> <p>3. Loadit, prog to load a file into memory until a reboot (Only the most esoteric hackers will find Loadit useful).</p> <p>Monolace CLI program resets Preferences to several colors of monochrome & interface screens. C source is included, works with DisplayPref, a CLI program which displays the current Preferences settings.</p> <p>BoingMachine A ray-traced animation of a perpetual motion Boing-making machine, includes the latest version of the Move program, which has the ability to play sounds along with the animation. By Ken Kifer</p> <p>Daisy Example of using the translator and narrator devices to make the Amiga talk. It is written in C.</p> <p>QuickFlx Script-driven animation and slideshow program flips through IFF images.</p> <p>BMon System monitor AmigaBasic program; perform simple manipulations of memory.</p> <p>Moose Random background program, a small window opens with a moose resembling Bulfinch saying wry phrases user definable.</p>
--	--	--	--

<p> DCGS Deluxe Grocery Construction Set, simple intuition-based prog for assembling and printing a grocery list. The Virus Check directory holds several programs relating to the software virus that came to the US from pirates in Europe as detailed in Amazing Computing V2.12. Bill Koester's full explanation of the virus code is included. One program checks for the software virus on a Workbench disk; the second program checks for the virus in memory, which could infect other disks. AMCUG Disk 25 Nemesis Graphics demo pans through space towards the mythical dark twin of the sun with wonderful music and space graphics. The KickPlay directory holds text that describes several patches to the Kickstart disk. For Amiga 1000 hackers who feel comfortable patching a disk in hexadecimal, KickPlay offers the chance to automatically do an ADOMEM for old expansion memory, as well as the ability to change the picture of the "Insert Workbench" hand. A program is also included for restoring the correct checksum of the Kickstart disk. KeyBrd BASIC prog edits keymaps, adjust the Workbench keymaps or create your own. Modifies the Workbench so three bitplanes are used, icons can have eight colors, instead of four, eight-color icons are included. Public domain program "zapicon" or "brush2icon" converts eight-color IFF brushes to icons, to use Deluxe Paint to make icons for this new Workbench. BrushIcon Converts brushes to icons (bitmap docs). Graphing prog reads [x,y] values from a file and displays them on the screen, similar to the same-named Unix program. Egraph Message-managing program for telecommunications, lets you save messages from an online transcript to another file, understands the message format of the national networks and several types of bulletin board software. Moves through the transcript and save messages. Keep 1.1 Speed up directory access, it creates a small file in each directory on a disk which contains the information about the files, will also remove all the "fastdir" files from each directory, by CLIMATE's authors. Kill.fastdir The LaseWB program changes between interface and non-interface Workbench. Previously, you were forced to reboot after changing Preferences to an interface screen. This program flips between the normal and extended screen heights. PW_Utility A shareware utility for ProWrite users, changes margin settings and font types. Guru A CLI program, prints out probable causes for Guru meditations; C source included. DiskWipe Latest from Software Distillery, removes files from directories or disk drives, much faster than "delete". Snow AmigaBasic makes snowflake designs. Mist Mailing list database. Sofballstats Maintain softball statistics/ team records. Dodge Short Module-2 program moves the Workbench screen around after a period of time, prevents monitor burn-in. AMCUG Disk 26 Todor Fay's SoundScape module code from his Amazing Computing articles. The source to Echo, Chord, TX, and VU is included. The Lattice and Manx C source code is here, along with the executable modules. ImageMaker Interesting tool edits image structures for C, loads & saves C code directly. Claz2 Update of prog to convert IFF images to PostScript files for printing on laser printers. SDBackup Hard disk backup prog with Lempel-Ziv compression to reduce the necessary number of disks. TCB Prints information about tasks and processes in the system; assembler source is included. FunBut Lets a function key act like a rapid series of left mouse button events. DC A handy program for people who use an Amiga 1020 5 1/4 inch drive as an AmigaDOS floppy. A Workbench program that sends a DiskChange signal to the operating system: instead of typing "diskchange df2:" over and over again, just click on the icon. C source included. System config File makes screen 80 columns wide of text in the Scribble word processor. Dick2Ram 2 programs to move the Scribble spelling dictionary to and from the RAM disk. Lexical Analyzes a text file and gives the Gunning-Fog, Flesch, and Kincaid indices which measure readability. HexDump Module-2 program to display memory locations in hexadecimal. Tartan AmigaBasic; design Tartan plaids. DrMaster Disk catalog program. BMP plays 8SVX sampled sounds in the background while something else is happening in the Amiga, as your Amiga is booting, for example. ShowPt CLI program changes your pointer to a given pointer. AMICUG 26 also has a collection of mouse pointers, & Workbench program to display them </p>	<p> Fred Fish Public Domain Software Fred Fish Disk 1: amigademo Graphical benchmark for comparing amigas. amigaterm simple communications program with Xmodem balls simulation of the "kinetic thingy" with balls on strings colorful Shows off use of hold-and-modify mode. dhystone Dhystone benchmark program. doty Source to the "doty window" demo on the Workbench disk. freedraw A small "paint" type program with lines, boxes, etc. gad John Draper's Gadget tutorial program ghmem Graphical memory usage display prog. halfrite demonstrates "Extra-Half-Byte" mode, if you have it hello simple window demo lastfp accessing the Motorola Fast Floating Point library from C palette Sample prog. to design color palettes. trackdisk Demonstrates use of the trackdisk driver. requesters John Draper's requester tutorial and example program. speech Sample speech demo program. speechy Stripped down "speechy". speechy2 Another speech demo program. Fred Fish Disk 2: alib Object module librarian. cc Unix-like frontend for Lattice C compiler. dbug Macro based C debugging package. make Machine independent. make2 Subset of Unix make command. microemcs Another make subset command. portar Small version of emacs editor, with macros, no extensions xrf Portable file archiver. Fred Fish Disk 3: gothic Gothic font banner printer. roff A "roff" type text formatter. ff A very fast text formatter clforth A highly portable forth implementation. Lots of goodies. xlisp Xlisp 1.4, not working correctly. Fred Fish Disk 4: banner Prints horizontal banner bgrep A Boyer-Moore grep-like utility bison GNU Unix replacement yacc, not working. bm Another Boyer-Moore grep-like utility grep DECUS C cross reference utility. kermi Simple portable Kermit with no connect mode. MyCLI Replacement CLI for the Amiga. V. 1.0 mandel A Mandelbrot set program, by Robert French and RJ Mical Fred Fish Disk 5: cons Console device demo program with supporting macro routines. freemap Creates a visual diagram of free memory inputdev sample input handler, traps key or mouse events joystick Shows how to set up the gameport device as a joystick. keyboard demonstrates direct communications with the keyboard. layers Shows use of the layers library mandelbrot IFF Mandelbrot program mouse hooks up mouse to right joystick port one.window console window demo parallel Demonstrates access to the parallel port printer opening and using the printer, does a screen dump, not working print.support Printer support routines, not working. proctest sample process creation code, not working region demos split drawing regions samplefont sample font with info on creating your own serial Demos the serial port singlePlayfield Creates 320 x 200 playfield speechy2 latest version of cute speech demo speechy simplified version of speechy, with IO requests text.demo displays available fonts timer demos timer device use trackdisk demos trackdisk driver Fred Fish Disk 6: compress like Unix compress, a file squeezer dad analog clock impersonator microemcs updated version of microemcs from disk 2 mult removes multiple occurring lines in files scales demos using sound and audio functions setparallel Allows changing parallel port parameters setserial Allows changing serial port parameters. sortc quicksort based sort program, in C strip Strips comments and extra whitespace from C source Fred Fish Disk 7: This disk contains the executables of the game Hack V 1.0.1. Fred Fish Disk 8: This disk contains the C source to Hack on disk 7. Fred Fish Disk 9: more Draws more patterns in black and white MVP-FORTH Mountain View Press Forth, version 1.00.03A. A shareware version of FORTH from Fantasia Systems. proff a more powerful text formatting program setface Prog to toggle interface mode on and off </p>	<p> skewb a rubik's cube type demo sparks moving snake Graphics demo Fred Fish Disk 10: conquest An interstellar adventure simulation game dehex convert a hex file to binary filezap Patch program for any type of file. fixobj Strip garbage off Xmodem transferred files. iff Routines to read and write iff format files. id simple directory program is Minimal UNIX is, with Unix-style wildcarding, in C squiq file squeeze and unsqueeze rek73 Star Trek game yacht Dice game. Fred Fish Disk 11: dpislide slide show program for displaying IFF images with miscellaneous pictures Fred Fish Disk 12: amiga3d Shows a rotating 3 dimensional solid "Amiga sign". ArgoTerm a terminal emulator program, written in assembler arrows3d Shows a rotating 3 dimensional wire frame arrow. ld4 directory listing program iconExec two progs for launching progs from Workbench, presently only works under CLI. SetWindow Makes an icon show a second image when clicked once SetAlternate terminal emulator, with ASCII Xmodem, dialer, more. StarTerm terminal emulator, with ASCII Xmodem, dialer, more. Fred Fish Disk 13: A Bundle of Basic programs, including: Jpad bybox xmodem addbook nr amiga-copy bounce box cardf colorcircles cube1 datedogstar dragon dynamictriangle Eliza fibuster fscape dart hal9000 hauntedM join loz mainpoint mouse pens pinwheel Readme rgb substage saletalk shuttle spaceort sketchpad sphere speechy striper spiral terminal termtest tm wheels xmodem note: some programs are Abasic, most are AmigaBasic, and some programs are presented in both languages) Fred Fish Disk 14: amiga3d update of #12, includes C source to a full hidden surface removal and 3D graphics beep Source for a function that generates a beep sound dex extracts text from within C source files dimensions demonstrates N dimensional graphics filezap update of disk 10, a file patch utility ghmem update of disk 1, graphic memory usage indicator gi converts IFF brush files to Image stuct, in C. pdterm simple ANSI VT100 terminal emulator, in 80 x 25 screen shell simple Unix "csh" style shell termcap mostly Unix compatible "termcap" implementation. Fred Fish Disk 15: Bobs graphics demo, like Unix "worms" Clock simple digital clock program for the tite bar Dazzle An eight-fold symmetry dazzler program. Fish Really pretty! double double buffered sequence cycle animation animation of a fish Monopoly A really nice monopoly game written in AbasC. OkiDataDump Okidata ML92 driver and Workbench screen dump program. Polydraw A drawing program written in AbasC. Polyfractals A fractal program written in AbasC. Fred Fish Disk 16: A complete copy of the latest developer IFF disk Fred Fish Disk 17: The NewTek Digi-View video digitizer HAM demo disk Fred Fish Disk 18: AmigaDisplay dumb terminal program with bell, selectable fonts Ash Perelease C Shell-like shell program, history, loops, etc. Browser wanders a file tree, displays files, all with the mouse MC68010 docs on upgrading your Amiga to use a MC68010 Multidm rotate an N dimensional cube with a joystick PgLatin SAY command that talks in Pg Latin Scripper Screen image printer Xisp1.6 source, docs, and execut for a Lisp interpreter. Fred Fish Disk 19: BackJack text-oriented blackjack game JayMiner Slides by Jay Miner, Amiga graphics chip designer, showing flowchart of the Amiga internals, in 640 x 400. Keymap_Test test program to test the key mapping routines LockMon Find unused file locks, for programs that don't clean up. </p>	<p> Fred Fish Disk 20: AmigaToAtari converts Amiga object code to Atari form DiskSalv program to recover files from a trashed AmigaDOS disk Hash example of the AmigaDOS disk hashing function Hd Hex dump utility aka Computer Language magazine, April 86 MandelBrot Mandelbrot contest winners Multitasking Tutorial and examples for Exec level multitasking Pack strips whitespace from C source PortHandler sample Port-Handler program that performs. Shows BCPL environment Random Random number generator in assembly, f or C or assembler. SetMouse2 sets the mouse port to right or left terminal SpeechTerm Emulator with speech capabilities, XModem TxEd Demo editor from Microsmith's Charlie Heath Fred Fish Disk 21: This is a copy of Thomas Wilcox's Mandelbrot Set Explorer disk. Very good! Fred Fish Disk 22: This disk contains two new "strains" of microemcs. Lemacs version 3.6 by Daniel Lawrence. For Unix V7, BSD 4.2, Amiga, MS-DOS, VMS. Uses Amiga function keys, status line, execute, startup files, more. Pemacs By Andy Poggio. New features include <ALT> keys as Meta keys, mouse support, higher priority, backup files, word wrap, function keys. Fred Fish Disk 23: Disk of source for MicroEmacs, several versions for most popular operating systems on micros and mainframes. For people who want to port MicroEmacs to their favorite machine. Fred Fish Disk 24: Conques interstellar adventure simulation game Csh update to shell on Disk 14, with built in commands, named variables substitution. Module-2 A pre-release version of the single pass Module-2 compiler originally developed for Macintosh at ETHZ. This code was transmitted to the AMIGA and is executed on the AMIGA using a special loader. Binary only. Fred Fish Disk 25: Graphic Hack A graphic version of the game on disks 7 and 8. This is the graphics-oriented game by John Toebes. Only the executable is present. Fred Fish Disk 26: UnHunk Processes the Amiga "hunk" loadfiles. Collect code, data, and bss hunks together, allows individual specification of code, data, and bss origins, and generates binary file with format reminiscent of Unix "a.out" format. The output file can be easily processed by a separate program to produce Motorola "S-records" suitable for downloading to PROM programmer. By Eric Black. C-kermi Port of the Kermit file transfer program and server. Ps Display and set process priorities Archi Yet another program for bundling up text files and mailing or posting them as a single file unit. Fred Fish Disk 27: ABDemos Amiga Basic demos; Carolyn Schepner. NewConvertFD creates .bmps from .id files. BPinsnes finds addresses of and writes to bitplanes of the screen's bitmap. AboutBmps A tutorial on creation and use of .bmps. LoadILBM loads and displays IFF ILBM pics. LoadACBM loads and displays ACBM pics. ScreenPcm creates a demo screen and dumps it to a graphic printer. Disassem Simple 68000 disassembler. Reads standard Amiga object files and disassembles the code sections. Data sections are dumped in hex. The actual disassembler routines are set up to be callable from a user prog so instructions in memory can be disassembled dynamically. By Bill Rogers. DvorakKeymap Example of a keymap structure for the Dvorak keyboard layout. Untested but included because assembly examples are few and far between. By Robert Burns Hypocycloids Sprograph, from Feb. 84 Byte. LinesDemo Example of proportional gadgets to scroll a SuperBMap. MemExpansion Schematics and directions for building your own homebrew 1 Mb memory expansion, by Michael Feilinger. SafeMalloc Program to debug "malloc()" calls ScienceDemos Convert Julian to solar and sidereal time, stellar positions and radial velocity epoch calculations and Galilean satellite plotter. By David Eagle. Fred Fish Disk 28: ABasic games by David Addison: Backgammon, Cribbage, Milestone, and Ornelo Cop DECUS 'cpc' C preprocessor, & a modified 'cc' that knows about the 'cpc', for Manx C. Shar Unix-compatible shell archiver, for packing files for travel. SuperBMap Example of using a ScrollLayer, syncing SuperBMaps for printing, and creating dummy RasPorts. Fred Fish Disk 29: AegisDraw Demo Demo program without save and no docs. Animator Demo Player for the Aegis Animator files </p>
--	--	--	--

Cc Enough	Unix-like frontend for Manx C Tests for existence of system resources, files, and devices	SendPacket	General purpose subroutine to send AmigaDOS packets.	Bom	Demo copy of B.E.S.T. Business Management System.	Fred Fish Disk 52	Assign	Replacement for AmigaDOS 'assign' command in C
Rubik	Animated Rubik's cube program	SpriteMaker	Sprite editor, can save work as C data structure. Shareware by Ray Larson.	BobList	A list of Amiga Bulletin Board Systems C compiler frontends for Manx and Latex C	Fractal	Fractal	Makes random fractal terrains
StringLib		Tracker	Converts any disk into files, for electronic transmission. Preserves entire file structure. Shareware by Brad Wilson.	Copper	A hardware copper list disassembler	Poly, HAMPoly	Poly, HAMPoly	Workbench-type demos for making polygons in lines and HAM
Vt100	Vt-100 terminal emulator with Kermit and Xmodem protocols	TriGops 3-D	space invasion game, formerly commercial, now public domain. From Geodesic Publications.	InstIFF	Converts instruments demo sounds to IFF sampled sounds	MaGads	MaGads	Example of mutual exclusion gadgets with GadgetText
Fred Fish Disk 30	Several shareware programs. The authors request a donation if you find their program useful, so they can write more software.	Unidelf	C preprocessor to remove given #ifdef sections of a file, leaving the rest alone. By Dave Yost	PopColours	Adjust RGB colors of any screen	Tek4010	Tek4010	Tektronix 4010 terminal emulator
BBS	an Amiga Basic BBS by Ewan Grantham	Vttest	VT-100 emulation test program. Requires a Unix system.	SpriteClock	Simple clock is displayed on a sprite above all screens	VDraw	VDraw	Paint-like drawing program
FinArt	Amiga art	Fred Fish Disk 36	Unix-like 'cp' copy program	ST Emulator	Non-serious Atari ST emulator	Fred Fish Disk 53	Animations	Demo animations with player program for Aegis Animator
FontEditor	edit fonts, by Tim Robinson	AcP	Updated version of clock on disk 15.	WBrun	Lets Workbench programs be run from the CLI	ARCre	ARCre	Creates rename scripts for files with long names, so they can be easily 'tacked and untacked'
MenuEditor	Create menus, save them as C source, by David Pehson	Cah	Manx 'cah'-like CLI history, variables, etc.	Wild	Two Unix shell style wild card matching routines	ARP	ARP	Preliminary AmigaDOS replacements for 'break', 'cd', 'chmod', 'echo', 'find', 'ls', 'mkdir', 'rm', 'rmdir', 'set', 'sleep', 'sort', 'tar', 'test', 'touch', 'unzip', 'wait', 'xargs', 'yes'
StarTerm3.0	Very nice telnet by J. Nangano	DierAid	Manx 'dier' like CLI history, variables, etc.	Fred Fish Disk 44	Miscellaneous icons	Compiler	Compiler	Not fully ported to the Amiga, this is a 68000 C compiler. It will produce simple assembly language output, but needs a lot of work.
(Fred Fish Disk 30 is free if requested when ordered with at least three other disks from the collection.)		Echo	Improved 'echo' command with color, cursor addressing	NewFF	New IFF material from CBM for sampled voice and music files	Spreadsheet	Spreadsheet	Update with source of the 'vt' spreadsheet on disk 36
Fred Fish Disk 31	Life game, uses blitter to do 19.8 generations a second.	FishLink	Fixes programs to let them run in external memory.	FullyTracePics	The famous ray-tracing pictures, from FFF38, now converted to IFF HAM for 'much' faster viewing.	TarSplit	TarSplit	Port of program to split Unix tar archives
Mandelbrot	Version 3.0 of Robert French's program.	Fm	Maps the sectors a file uses on the disk.	ViewLBM	Displays normal and HAM LBM files	UUnencode	UUnencode	Utilities to encode and decode binary files for ASCII transmission, expanding them by 35%
MaxExample	Mutual exclusion gadget example.	KickBench	Docs, program to make a work disk that works like a Kickstart and Workbench.	Fred Fish Disk 45	Icons	Fred Fish Disk 54	Hanoi	Solves Towers of Hanoi Problem in it's own Workbench window, by Ali Ozer
RainSpeed	Measure relative RAM speed, chip and fast.	Lex	Computes Fog, Flesch, and Kincaid readability of text files.	Icons	Miscellaneous icons	ISpell	ISpell	Port of a Unix screen oriented, interactive spelling checker. (Expansion RAM required) by Pace Willisson
Set	Replacement for the Manx 'set' command for environment variables, with improvements.	TunnelVision	David Addison ABasic 3D maze perspective game.	Make	Another 'make', with more features	Ing	Ing	A Screen of lots of bouncing little windows by Leo 'Bois Ewhad' Schwab
Tree	Draws a recursive tree, green leafy type, not files.	Vc	Version 2.2 of Dave Wecker's telnet program	Pictures	Miscellaneous pictures	Lav	Lav	Displays number of tasks in run queue, averaged over last 1, 5, and 15 minute periods. by William Rucklidge
TxEd	Crippled demo version of Microsmith's text editor, TxEd.	Vt100	Version 2.2 of Dave Wecker's telnet program	Update	Updates an older disk with newer files from another disk	MIDITools	MIDITools	Programs to play/record through the MIDI I/F. by Fred Cassiner
VDraw	Full-featured drawing program by Stephen Vermeulen.	YaBoing	Onig' style game program shows sprite collision detects	WhereIs	Searches a disk for files of given name	MoreRows	MoreRows	Program to make the Work Bench Screen larger than normal, by Neil Katin and Jim MacKraz
Xcon	Invokes CLI scripts from icon	Fred Fish Disk 37	This disk is a port of Timothy Budd's Little Smaltalk system, done by Bill Kinnerley at Washington State University.	Fred Fish Disk 46	Shareware 68010 macro assembler, ROM Kernel Manual compatible	Tit	Tit	Program to make your Amiga look like it didn't pass vibration testing. by Leo 'Bois Ewhad' Schwab
Tcon	Displays text files from an icon.	Fred Fish Disk 38	CSquared Sep 86 Sci American, Circle Squared algorithm	Aem	Shareware 68010 macro assembler, ROM Kernel Manual compatible	Fred Fish Disk 55	Can	V2.05 of Matt Dillon's can like shell (Modified for Manx C), by Matt Dillon
Fred Fish Disk 32	Extended address book, AmigaBASIC	FixObj	Strips garbage off Xmodem transferred object files	CheckModem	'execute' file program detects presence of modem	NewStartups	NewStartups	New C Startup modules: by Andy Finkle
Calendar	Calendar/diary program, AmigaBASIC	Handler	AmigaDOS handler (device) example from C-A	Egad	Gadget editor from the Programmers Network	AStartup.asm	ASStartup.asm	Opens a startup window, using user specs. by Commodore
DotPlus1	First volume of CLI oriented developer tools	Hp-10c	Mimics a HP-10C calculator, written in Module-2	Jive	Transforms a file from English to Jive.	TWSStartup.asm	TWSStartup.asm	Opens a startup window, using user specs. by Commodore
DotPlus2	2nd volume of CLI oriented developer tools	IFFEncode	Saves the screen as an IFF file	MyLib	A binary only copy of Ma's alternate runtime library. Author: Matt Dillon	Palette	Palette	Change another program's screen colors. by Carolyn Scheppe
Executables only:		IFFDecode	Dumps info about an IFF file	ProfMacros	Subset Berkeley 'ms' and 'tm' macros for 'prof'	PipeDevice	PipeDevice	Allows the standard output of one process to be fed to the standard input of another. by Matt Dillon
MacView	Views MacPaint pics in Amiga low or high res, no sample pictures, by Scott Evernden.	IFFDump	Dumps info about an IFF file	ValSpeak	Transforms a file from English to Valley Speak.	ScreenSave	ScreenSave	Saves a normal or HAM mode screen as an IFF file. by Carolyn Scheppe
Puzzle	Simulation of puzzle with moving squares.	Jah	BOS C-like CLI shell	Fred Fish Disk 47	Simulation of a robotic arm, very good graphics, teaching tool, including C source.	ShanghaiDemo	ShanghaiDemo	Demo of the Activation game Shanghai.
ShowHAM	View HAM pictures from CLI.	NewStat	STATUS-like program, shows priority, processes	3D-Arm	Simulation of a robotic arm, very good graphics, teaching tool, including C source.	SoundExample	SoundExample	A double buffered sound example for Manx C. by Jim Goodnow
Solitaire	ABasic Games of Canfield and Klondike, from David Addison.	Reversi	Game of Reversi, version 5.1	Juggler	Eric Graham's stunning HAM animation of a robot juggler	Vaprites	Vaprites	A working vaprite example, by Eric Cotton
Spir3	Graphics demo of spinning cubes, double-buffered example.	Ucodecode	Translate binary files to text, Unix-like programs	VT-100	Version 2.4 of Dave Wecker's terminal emulator, with Xmodem and Kermit file transfer protocols	Vt100	Vt100	V2.6 of Dave's Vt100 terminal emulator with kermit and xmodem, by Dave Wecker
Sword	Sword of Fallen Angel text adventure game written in Amiga Basic.	Vdraw	Drawing program, version 1.14	Fred Fish Disk 48	Alpha version of a hard disk file archiver	Fred Fish Disk 56	Clipboard	Clipboard device interface routines, to provide a standard interface, by Andy Finkle
Trails	Leaves a trail behind mouse, in Module-2	VoiceFilter	DX MIDI synthesizer voice filter program	Cyclods	Update of electronic spirograph from disk 27	Clipboard	Clipboard	Clipboard device interface routines, to provide a standard interface, by Andy Finkle
Fred Fish Disk 33	3d version of the 'stars' program below.	Window	Example of creating a DOS window on a custom screen	DrUtil	Enhanced version of DrUtil from disk 35	ConPackets	ConPackets	Demo of use of DOS Packets, ConUtil, etc. by Carolyn Scheppe
3dstars	Low-level graphics example scrolls bitmap with ScrollVPort	Fred Fish Disk 39	'echo', 'touch', 'ls', 'ls' written in assembler.	MyUpdate	Disk update utility with options for stripping comments from C header files, and interactive verification of the updating process	GetDisks	GetDisks	Program to find all available disk device names and return them as an exelist, by Philip Lindsay
Bginap	Double-buffered animation example for BOBs and VSPorts.	AnsEcho	Displays HAM images from a ray-tracing program, with sample pictures.	Plot	Computes and displays 3 dimensional functions in hires	GetVolume	GetVolume	Program to get volume name of the volume that a given file resides on. by Chuck McManis
DoubleGels	Double-buffered animation example for BOBs and VSPorts.	Display	Example device driver source, acts like RAM disk	Polygon	Moire type pattern generator with color cycling	Icon2C	Icon2C	Reads an icon file and writes out a fragment of C code with the icon data structures. by Carolyn Scheppe
DisMapper	Displays sector allocation of floppy disks.	Driver	Example device driver source, acts like RAM disk	OMouse	Moire type pattern generator with color cycling	MergeMem	MergeMem	Program to merge the MemList entries of sequentially configured RAM boards. by Carolyn Scheppe
MainView	View memory in real time, move with joystick.	Xisp	Xisp 1.7, executable only	Touch	Example of setting the timestamp on a file, using a technique from Commodore-Amiga	mCAD	mCAD	An object oriented drawing program, V1.1 by Tim Mooney
Oing	Bouncing balls demo	Fred Fish Disk 40	Terminal emulator with Xmodem, Kermit and CIS B protocols, function keys, scripts, RLE graphics and conference mode.	Trees	More extensive version of the trees program on disk 31	Fred Fish Disk 57	Replaced by FF97 Due to Copyright problems	
Spring	Oing, with sound effects.	AmigaMonitor	Dynamically displays the machine state, such as open files, active tasks, resources, device states, interrupts, libraries, ports, etc.	BreakOut	A brick breakout game, uses 3-D glasses	Fred Fish Disk 58	ASDG-rtd	Extremely useful shareware recoverable ram disk, by Perry Kivolowitz
ScreenDump	Dumps highest screen or window to the printer.	Arc	Popular file compression system, the standard for 'transferring' files	DiskZap	Version 1.1 of a program to edit disks and binary files	BgView	BgView	Displays any IFF picture, independent of the physical display size, using hardware scroll. by John Hodgson
Sdt	Simple database program from a DECUS tape.	AreaCode	'aink' replacement linker, version 6.5	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands	EGraph	EGraph	Reads pairs of x and y values from a list of files and draws a formatted graph. by Lawrence Turner
Stars	Star field demo, like Star Trek.	Blink	An 'asteroids' clone.	Missile	A Missile Command-type game, with sound, in assembler	HyperBase	HyperBase	Shareware data management system. V1.5
StarTermPlus	Terminal program with capture, library, function keys, Xmodem, CS-B protocols.	Cosmo	Data General D-210 Terminal emulator	PerfectSound	Sound editor for a low-cost sound digitizer	MemClear	MemClear	Walks through the free memory lists, zeroing free memory along the way. by John Hodgson
Vt100	Version 2.0 of Dave Wecker's VT-100 emulator, with scripts & function	Dg210	Windowed DOS interface program, V 1.4	Sizzlers	Graphics demos	NewZAP	NewZAP	A third-generation multi-purpose file sector editing utility. V3.0 by John Hodgson
Fred Fish Disk 34	Support files for Gimpel's 'int' syntax checker	DrUtil	Windowed AmigaDOS CLI help program	UnixArc	Ver 'of' for Unix System V machines, in C	RainBow	RainBow	A Mauser-style rainbow generator. by John Hodgson
Alint	PO 'alink' compatible linker, faster, better.	PagePrint	Prints text files with headers, page breaks, line numbers	Wombat	Version 3.01 of Dave Walker's terminal emulator	SMUSPlayers	SMUSPlayers	Two SMUS forms, to play SMUS IFF music formatted files. by John Hodgson
Blink	Updated to FF 18 browser, in Manx, with scroll bars, bug fixes.	PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4	View	View	A tiny LBM viewer by John Hodgson
Browser	Updated to FF 18 browser, in Manx, with scroll bars, bug fixes.	SpriteEd	Sprite Editor edits two sprites at a time	Fred Fish Disk 42	Update to the file compression program on disk 6	WBump	WBump	JX-80 optimized workbench printer that does not use DumpRPort, by John Hodgson
Btree	b-tree data structure examples	X-Spell	Spelling checker allows edits to files	Fred Fish Disk 43	Update to the file compression program on disk 6			
Btree2	Another version of 'btree'	AmigaVenture	Create your own text adventure programs in AmigaBasic	BreakOut	A brick breakout game, uses 3-D glasses			
Calendar	Appointment calendar with alarm.	Cah	Version 2.03 of Dillon's C-sh-like shell.	DiskZap	Version 1.1 of a program to edit disks and binary files			
Less	File viewer, searching, position by percent, line number.	Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
NewFonts	Set of 28 new Amiga fonts from Bill Fischer	DrUtil	Windowed DOS interface program, V 1.4	Missile	A Missile Command-type game, with sound, in assembler			
Pr	Background print utility, style options, wildcards.	DOSHelper	Prints text files with headers, page breaks, line numbers	PerfectSound	Sound editor for a low-cost sound digitizer			
Requester	Deluxe Paint-type file requester, with sample.	PagePrint	Prints text files with headers, page breaks, line numbers	Sizzlers	Graphics demos			
Fred Fish Disk 35	C example of making asynchronous I/O calls to a DOS handler, written by C-A	PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	UnixArc	Ver 'of' for Unix System V machines, in C			
ASandPacket	C example of getting the intuition pointer a CON or RAW: window, for 1,2, by C-A.	SpriteEd	Sprite Editor edits two sprites at a time	Wombat	Version 3.01 of Dave Walker's terminal emulator			
ConsoleWindow	C example of getting the intuition pointer a CON or RAW: window, for 1,2, by C-A.	X-Spell	Spelling checker allows edits to files	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
DirUtil	Walk the directory tree, do CLI operations from menus	AmigaVenture	Create your own text adventure programs in AmigaBasic	Fred Fish Disk 42	Update to the file compression program on disk 6			
DirUtil2	Another variant of DrUtil	Cah	Version 2.03 of Dillon's C-sh-like shell.	Fred Fish Disk 43	Update to the file compression program on disk 6			
FileRequester	Latex C file requester module, with demo driver, from Charlie Heath.	Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	BreakOut	A brick breakout game, uses 3-D glasses			
MacView	Views MacPaint pictures in Amiga low or high res, with sample pictures, by Scott Evernden.	DrUtil	Windowed DOS interface program, V 1.4	DiskZap	Version 1.1 of a program to edit disks and binary files			
PloP	Simple IFF reader program	DOSHelper	Prints text files with headers, page breaks, line numbers	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
PopCLI	Sidekick-style program invokes a new CLI, with automatic screen blanking.	PagePrint	Prints text files with headers, page breaks, line numbers	Missile	A Missile Command-type game, with sound, in assembler			
QuickCopy	Devonport disk copiers duplicate copy-protected disks.	PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	PerfectSound	Sound editor for a low-cost sound digitizer			
ScrollPi	Dual playfield example, from C-A, shows 400 x 300 x 2 bit plane playfield on a 320 x 200 x 2 plane deep playfield.	SpriteEd	Sprite Editor edits two sprites at a time	Sizzlers	Graphics demos			
		X-Spell	Spelling checker allows edits to files	UnixArc	Ver 'of' for Unix System V machines, in C			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Fred Fish Disk 42	Update to the file compression program on disk 6			
		DrUtil	Windowed DOS interface program, V 1.4	Fred Fish Disk 43	Update to the file compression program on disk 6			
		DOSHelper	Prints text files with headers, page breaks, line numbers	BreakOut	A brick breakout game, uses 3-D glasses			
		PagePrint	Prints text files with headers, page breaks, line numbers	DiskZap	Version 1.1 of a program to edit disks and binary files			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		SpriteEd	Sprite Editor edits two sprites at a time	Missile	A Missile Command-type game, with sound, in assembler			
		X-Spell	Spelling checker allows edits to files	PerfectSound	Sound editor for a low-cost sound digitizer			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	Sizzlers	Graphics demos			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	UnixArc	Ver 'of' for Unix System V machines, in C			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		DrUtil	Windowed DOS interface program, V 1.4	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		DOSHelper	Prints text files with headers, page breaks, line numbers	Fred Fish Disk 42	Update to the file compression program on disk 6			
		PagePrint	Prints text files with headers, page breaks, line numbers	Fred Fish Disk 43	Update to the file compression program on disk 6			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	BreakOut	A brick breakout game, uses 3-D glasses			
		SpriteEd	Sprite Editor edits two sprites at a time	DiskZap	Version 1.1 of a program to edit disks and binary files			
		X-Spell	Spelling checker allows edits to files	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	Missile	A Missile Command-type game, with sound, in assembler			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	PerfectSound	Sound editor for a low-cost sound digitizer			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Sizzlers	Graphics demos			
		DrUtil	Windowed DOS interface program, V 1.4	UnixArc	Ver 'of' for Unix System V machines, in C			
		DOSHelper	Prints text files with headers, page breaks, line numbers	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		PagePrint	Prints text files with headers, page breaks, line numbers	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	Fred Fish Disk 42	Update to the file compression program on disk 6			
		SpriteEd	Sprite Editor edits two sprites at a time	Fred Fish Disk 43	Update to the file compression program on disk 6			
		X-Spell	Spelling checker allows edits to files	BreakOut	A brick breakout game, uses 3-D glasses			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	DiskZap	Version 1.1 of a program to edit disks and binary files			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Missile	A Missile Command-type game, with sound, in assembler			
		DrUtil	Windowed DOS interface program, V 1.4	PerfectSound	Sound editor for a low-cost sound digitizer			
		DOSHelper	Prints text files with headers, page breaks, line numbers	Sizzlers	Graphics demos			
		PagePrint	Prints text files with headers, page breaks, line numbers	UnixArc	Ver 'of' for Unix System V machines, in C			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		SpriteEd	Sprite Editor edits two sprites at a time	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		X-Spell	Spelling checker allows edits to files	Fred Fish Disk 42	Update to the file compression program on disk 6			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	Fred Fish Disk 43	Update to the file compression program on disk 6			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	BreakOut	A brick breakout game, uses 3-D glasses			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	DiskZap	Version 1.1 of a program to edit disks and binary files			
		DrUtil	Windowed DOS interface program, V 1.4	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		DOSHelper	Prints text files with headers, page breaks, line numbers	Missile	A Missile Command-type game, with sound, in assembler			
		PagePrint	Prints text files with headers, page breaks, line numbers	PerfectSound	Sound editor for a low-cost sound digitizer			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	Sizzlers	Graphics demos			
		SpriteEd	Sprite Editor edits two sprites at a time	UnixArc	Ver 'of' for Unix System V machines, in C			
		X-Spell	Spelling checker allows edits to files	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	Fred Fish Disk 42	Update to the file compression program on disk 6			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Fred Fish Disk 43	Update to the file compression program on disk 6			
		DrUtil	Windowed DOS interface program, V 1.4	BreakOut	A brick breakout game, uses 3-D glasses			
		DOSHelper	Prints text files with headers, page breaks, line numbers	DiskZap	Version 1.1 of a program to edit disks and binary files			
		PagePrint	Prints text files with headers, page breaks, line numbers	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	Missile	A Missile Command-type game, with sound, in assembler			
		SpriteEd	Sprite Editor edits two sprites at a time	PerfectSound	Sound editor for a low-cost sound digitizer			
		X-Spell	Spelling checker allows edits to files	Sizzlers	Graphics demos			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	UnixArc	Ver 'of' for Unix System V machines, in C			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	Wombat	Version 3.01 of Dave Walker's terminal emulator			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	Fred Fish Disk 41	GNU for Unix 'yacc', working update to FF4			
		DrUtil	Windowed DOS interface program, V 1.4	Fred Fish Disk 42	Update to the file compression program on disk 6			
		DOSHelper	Prints text files with headers, page breaks, line numbers	Fred Fish Disk 43	Update to the file compression program on disk 6			
		PagePrint	Prints text files with headers, page breaks, line numbers	BreakOut	A brick breakout game, uses 3-D glasses			
		PopCLI	Starts a new CLI with a single keystroke, from any program. With a screen-saver feature. Version 2.0 with Sprite Editor edits two sprites at a time	DiskZap	Version 1.1 of a program to edit disks and binary files			
		SpriteEd	Sprite Editor edits two sprites at a time	FirstSilicon	A smart CLI replacement with full editing and recall of previous commands			
		X-Spell	Spelling checker allows edits to files	Missile	A Missile Command-type game, with sound, in assembler			
		AmigaVenture	Create your own text adventure programs in AmigaBasic	PerfectSound	Sound editor for a low-cost sound digitizer			
		Cah	Version 2.03 of Dillon's C-sh-like shell.	Sizzlers	Graphics demos			
		Doug	Macro based C debugging package, FF #2 example from CBM, update to intuition manual	UnixArc	Ver 'of' for Unix System V machines, in C			

Fred Fish Disk 58 Browser Update to browser program on disks 18 and 34. S-E Browser2 Another different browser program. E Clock Clock program with fonts, colors. E Dme Dime text editor V1.22 for programmers. ED DropCloth Puts pattern on Workbench backdrop. ED DropShadow Puts shadows on Workbench windows. E-D FixWB Similar to DropCloth, but doesn't work yet. S-D mCAD Object-oriented drawing program, version 1.2.2. Much improved over disk 56. Robotoff Demo of animated pointers on Workbench. S-E-D Supermort General compounding/amortization loan calculator. E-D	Fred Fish Disk 68 Am58k Macro assembler, v1.0.3, E-D BitLab Bitwise exploring program, in C, S-E-D Conman Replacement console device handler adds editing and history to any application that uses CON. v0.8, E-D Console Replacement console routines, in C, S-E-D Dk Deays the screen bit by bit, update to disk 66, in Modula-2, S-E-D Frags Displays memory fragmentation by listing the size of free memory blocks, in C, S-E-D IconType Change the type of an icon, in C, S-E-D Make 'make' in Manx C, S-E-D MonProc Monitors processes for packet activity, in C, S-E-D MouseClock Mouse pointer into a digital clock, in C, S-E-D Sb Browses system structures, from Transactor magazine, v1.0, in C, S-E-D Spew Generates National Enquirer-type headlines from rules file, in C, S-E-D Spool Three programs to demonstrate multitasking and spooling in a printer spooler. In C, v1.2, S-E-D Wc Counts words ala Unix 'wc', but faster, in C, S-E-D	DropShadow WB dropshadows, v2.0. Update FF59. E-D Funds AmigaBASIC prog tracks mutual or stocks-D Less Text viewing program, like Unix 'more', v1.1, update to disk 34. S-E-D Makemake Scans C source files and constructs a vanilla 'makefile' in the current directory. S-E-D mCAD Object-oriented drawing prog, v1.2.4, update to FF 59. Shareware, E-D Random Simple random number generator in C. S-E-D TDebug Monitors devices by intercepting Exec SendIO and DoIO vectors, in C, v1.0, S-E-D Units Converts measurements in different units, includes 'chart' option, in C, S-E-D XCopy Replacement for AmigaDOS 'copy', doesn't change the date, uses Unix wildcards. E-D	Ed Simple editor, similar to Unix 'ed', based on the editor in Software Tools. GravityWars Game of planets, ships and black holes, v1.04, update to disk 70. HunkPad Adds legal padding to executables for Xmodem transmission. PipeHandler An AmigaDOS pipe device which supports named pipes and taps. V1.2 PopCLI V3.0 of a hot-key to invoke a CLI window, with screen blanker, update to disk 40. Requester Update FF34, file requester similar to DPaint. ScottDevice V3.1 of a 'mountable' MicroForge SCSI driver. Viacom Another Schwab hack, makes TV-like static on screen. Parody
Fred Fish Disk 60 Various shareware and freeware programs Bitz Memory resident file viewer. Very fast. E-D BitzFonts Makes text output faster. E-D BitzShake Terminal emulator with VT52/VT100/VT102 support. E-D Med Mouse-driven text editor for version 2.1. E-D PrtDrvGen Generates printer drivers, version 1.1.5 available from author. E-D Show Slideshow-like IFF viewer, V2.1. E-D Uedit Customizable text editor V2.0. E-D Ueturbio Example Uedit setup macros. S-E-D	Fred Fish Disk 70 This is a disk of shareware programs. AmigaMonitor Explores state of the system, v1.13 Arc Standard file compressor and librarian, v2.23, a port of MS-DOS v5.0. E-D BlackBook Phone book program. DoTli Intuition-driven file manipulator program, v2.0. GravityWars Game of planets, ships and black holes, v1.03. Jobs Alternate user interface to CLI and WB, v2.1. Lens Magnifies area around mouse, shows it in a window, v1.0. Life-3d 3D version of the classic cellular-automaton game, v1.2. Logo Logo language interpreter SetKey Demo keypad editor, v1.0 Vpg Makes displays for aligning video monitors, v1.0	Fred Fish Disk 75 Bezier Play with Bezier curves points and granularity. S-E-D BSplines Play with b-splines, as above, S-E-D Comm C source for Comm terminal program v1.34. S-E-D Copy Replacement 'copy' command v1.0, preserves date, in C, S-E-D Diff Simple diff in C, S-E-D DuM2 Another Diff in Modula-2, v1.5, S-E-D Eless Fast 'diff' program in C, S-E-D Fd Faster 'less' in C, S-E-D HardCopy Sends a transcript of a CLI session to a file, in C, S-E-D MouseOff Update FF73, turns off mouse pointer, S-E-D SelfFont Changes the font in a Workbench screen, v2.0, S-E-D SpeedDir Another fast 'dir', in assembler, S-E-D	Fred Fish Disk 85 Csh V2.06 of Dillon's 'csh'-like shell FileReq Source to wildcard file requester Hide Hides expansion memory from programs ImageTools Shareware tools to manipulate IFF images LowMem Server/Shared library to aid in low memory situations Plot6 A star plotting program with source. RawIO Example of setting raw mode on standard input Rocket Lunar Lander for Workbench, with source. Vmore 'more'-like text viewing utility, v1.0 SE Vnews Simple Unix news reader. Fred Fish Disk 86 AutoPaintAuto-selects window under the mouse pointer, with screensaver. ClickToFront Double-clicks in window brings it to front, v1.1, S-E-D Cmd V3.0 of a tool to redirect printer output to a file. FileISG-Demo Demo of Software File Ilog, a database manager with sound and graphics.
Fred Fish Disk 81 ATPatch Patches Transformer to work under AmigaDOS 1.2. S-E-D FillDisk Writes zeroes to free blocks on a disk for security. S-E-D LPatch Patch for programs that abort when loading under AmigaDOS 1.2. S-E-D MicroEmacs Conroy MicroEmacs V3.8b, newer than disk 22. S-E-D PearFont Like Topaz, but rounded edges. Terrain Generates fractal scenery. S-E-D Vspires Makes 28 Vspires, from S-PatchBook.	Fred Fish Disk 82 This is a port of the Unix game 'Hack', by the Software Distillery, version 10.30. Fred Fish Disk 83 This is a port of the Unix game 'Larn', by the Software Distillery, version 12.08.	Fred Fish Disk 87 AdvSys Adventure system from Byte May 1987, v1.2 E-D AutIconOpen Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D Claz Converts IFF files to PostScript, V2.0, SED Commodi Commodities Exchange, an exact library to manage input handler, v0.4 Diff Update to disk 75 of Unix-like 'diff', S-E-D Dme V1.27 of Dillon's text editor, update FF74, E-D DropShadow V2.0 of program that puts shadows on Workbench, S-E-D	Fred Fish Disk 87 AdvSys Adventure system from Byte May 1987, v1.2 E-D AutIconOpen Fools Workbench to open disk icons, V1.2 update to disk 73, S-E-D Claz Converts IFF files to PostScript, V2.0, SED Commodi Commodities Exchange, an exact library to manage input handler, v0.4 Diff Update to disk 75 of Unix-like 'diff', S-E-D Dme V1.27 of Dillon's text editor, update FF74, E-D DropShadow V2.0 of program that puts shadows on Workbench, S-E-D
Fred Fish Disk 85 Bawk Unix text processor, like 'awk'. Doesn't work, but source is included. S-E-D MWB Example of rerouting Workbench window open calls to another custom screen. Version 1.01, S-E-D GoseWB Example for dosing a custom Workbench screen. S-E-D Cookie Generates one-line fortune-cookie aphorisms. S-E-D JTime Build-your-own mouse port doc. MenuBuilder Creates C source files for menus, based on text descriptions. S-E-D NewPackets CBM tutorial on new packets and structures in AmigaDOS 1.2 PascalToC Pascal to C translator, not so great. S-E-D Prep 'tutor'-like FORTRAN preprocessor. S-E-D RunBack Creates and animates 3-D objects, v0.5, E-D SunMouse This program automatically clicks in windows when the mouse is moved over them. Version 1.0, E-D	Fred Fish Disk 86 Add Customizes existing program menus with Amiga-key shortcuts. Also includes 'unli', which waits until a given window is created. Shareware, in C, S-E-D AutoIconOpen Fools WB into thinking mouse has double-clicked icons. In C, S-E-D Dio Generic Exec device interface code for opening libraries, getting multiple I/O channels, asynchronous operations, etc. In C, S-E-D Dissolve Slowly displays IFF files, ala Nov 86 Dr. Dobbs' program. In C, S-E-D DTerm Flexible, reprogrammable terminal program v1.10, E-D Expose Re-arranges windows so that at least one pixel of menu bar gadgets are exposed. In C, S-E-D Lit Scans a text file, converts to C-style printable strings, v2.0, S-E-D Lmv 'Long Movie', program views series of IFF pics in quick succession, upto 18 fps. Shareware, E-D MouseOff Mouse pointer disappears after ten seconds of non-use. In C, S-E-D ParOut Examples of controlling parallel port with resources instead of the PAR device. In C, S-E-D	Fred Fish Disk 87 AssignDev Give devices multiple names, in C, S-E-D AuxHandler Example of a dos handler that allows use of a CLI via the serial port. Includes source. Author: Steve Drew Cmd Redirects printer output to a file, in C, S-E-D Info AmigaDOS 'info' replacement, in C and assembler, S-E-D Kill Removes a task and its resources, in C, S-E-D MZError Displays errors from TDI Modula-2 compiles, S-E-D MonProc Update to process packet program from disk 68, in C, S-E-D Mounted Program for testing if a drive is present, in a script In C, S-E-D Nro Another 'roff'-style text formatter, in C, S-E-D ParTask Finds parent task, in C, S-E-D QueryAny For scripts, asks a question, accepts Y/N, gives return code. In assembler, S-E-D SetSize Resets pref settings for screen size, in C, SED SharedLib Example, shared lib, in C & assembler, S-E-D Task Simple CreaTask() example in C, S-E-D Uw Unix Windows client v1.0, in C, S-E-D Who Lists tasks on ready and wait queues, in C, S-E-D	Fred Fish Disk 88 FF Disk 88 has been removed due to copyright problems Fred Fish Disk 89 DiskCatalogue Disk catalogue program, V1.0a, E-D Funckey Shareware function key editor, V1.01, E-D MFF-Demo Demo of MicroFiche Filer database prog ScreenShift Adjust screen position in Preferences, SED Snake Bouncing squiggly lines demo, S-E-D AutoEnginier screen contraction requester improvement S-E-D Demolition Display Hack S-E-D Fred Fish Disk 90 (replaces Fred Fish 80) AmGazer Night sky viewer of 1573 stars, set date, time, day. E-D CardFile AmigaBasic card file study aid. E-D Conman Console handler replacement gives line editing and history to most progs, v0.98, E-D IMandelbrot Slight update to disk 78 Mandelbrot program, E-D NewDemos Replacements for lines and boxes demos that take less CPU time, E-D Othello Game of Othello, E-D PrintText Displays text files with gadgets, speech, IFF display, v1.2, E-D PrtDrvGen Automatic printer drv. generator, v2.2b, ED RainBench Cycles colors of WB backdrop or text. ED ShortOut Makes single-key shortcuts for entering commonly typed CLI commands, & custom macros. E-D ShowPrint Displays and prints all sizes of IFF pictures & controls printer output styles, v2.0 E-D Sizzlers Small Workbench timer counts time and \$/minute. E-D Tools I/Ovectors tools: a memory editor, memory disassembler, ASCII chart, and calculator. E
Fred Fish Disk 88 AmSci Preliminary plans for a SCSI disk controller board. Am58k Macro assembler, version 1.0.1. E-D Assigned Example for avoiding DOS insert-disk requester, by scanning the list of 'assigned' names. S-E-D Dk Pretends to eat away at CLI window. S-E-D Flip Flips whole screen as a joke. S-E-D Foogol Foogol cross-compiler generates VAX assembly code. S-E-D Free Prints amount of free space on all drives. S-E-D MalloTest malloc/free memory test program. S-E-D Melt Pretends to melt the screen. S-E-D Nart Graphic flying string demo. S-E-D Purty Easy way to set printer attributes from Workbench. E-D RayTracer Simple ray tracing program. E-D SendPackets Updated CBM examples of packet routines on disk 35. S-E-D SnapShot Memory resident screen dump. E-D TapBBS Shareware BBS system, version 1.02. Fred Fish Disk 87 AmCat Shareware disk catalog program. AmigaSpell Shareware Intuition spelling checker, V2.0. E-D Bouncer 3-D bouncing ball written in MultiForth, SED Comm Terminal program version 1.33, E-D Dux5 Another version of DiUHL. S-E-D HexCalc Hex, octal, & decimal calculator. E-D Icons Various big and alternate image icons. Mandala Mandala graphics and sound. E PersMail Demo shareware personal file manager. RSLClock Menu bar clock version 1.3. E-D RTCubes Graphics demo of 3D cubes. E-D Wheel 'Wheel of Fortune'-type game, AmigaBASIC Fred Fish Disk 90 This is version MG 1b of the MicroGNUMacs. Source and executable are included, as well as source for other computers besides the Amiga.	Fred Fish Disk 74 Gled Edits and recalls CLI commands, v1.3, E-D Control Interprets graphic printer dump calls and accesses color map, wdr, and screen resolution. C, S-E-D Dme Simple WYSIWYG text editor for programmers, v1.25. Update of FF 59, E-D	Fred Fish Disk 89 This disk contains a demo version of TeX from N Squared. It is limited to small files, and the previewer can only display ten pages or less, and only a small number of fonts are provided. Fred Fish Disk 91 AudioToolsPrograms from Rob Peck's July/August Amiga World article BitLab Bitwise experimentation program, V1.2, update to FF68	Fred Fish Disk 91 Adventure Definition Language (ADL) a superset of an older language called DDL by Michael Urban, Chris Kostasnick, Michael Stein, Bruce Adler, and Warren Usui. ADL enhancements by Ross Cunliffe. Included are sources to the ADL compiler, interpreter, and debugger. Binaries combined by Ross with Latex 3.03. CLI environment only. Documentation is available from the authors.

Fred Fish Disk #2	Spines	Prog to demonstrate curve fitting & rendering techniques. by Helene (Lee) Taran	IP20C	A util. to write a C-lang definition to mimic the intention pointer.S	V100	V2.7 of vt100 terminal emulator with kermit & xmodem file transfer. Includes a few bug fixes posted to Usenet shortly after the posting of v2.7. Update to FF55. Includes S. By Dave Wecker
As502	Shm	Simple graphics demo, approximately simulates the motion of two interacting pendulums. Includes Source by Chris Ediss	Pere-et-Fil	Ext. of creating & using reentrant processes. S. Similar to 'Journal' v2.0 update to FF95.	WBlander	A special version of the WBlander program from FF100. Ending is unique. Effective use of sound, Includes S. By Peter da Silva & Karl Lehenbauer
Bawk	Fred Fish Disk #3	16 color terminal program based on Comm V1.34. Includes Macro window, custom gadgets, colorized menus, etc. V. Beta 0.18 by Keith Young, comm by D.J. James. E. Writes AmigaDOS disks as the backup destination. recovers files from the backup disk. Requires manual decisions on disk structure. by Alan Kent SE	Fred Fish Disk 106	Shareware function key editor, v1. update to FF89. Source avail. from author (Anson Mah).	Fred Fish Disk 115	Masterful Video commercial of the Amiga, Beatles music, requires one meg of memory to run. Binary only. By Robert Wilt
HunkPad	Access	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	MoreArt	A small selection of some Amiga artwork.	Marketoid	Another deviant sprite oriented demo with lots of 'in' jokes. 512K required, includes S. By Leo Schwab
Less	Backup	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	QuickFix	An IFF slideshow and cel animation prog v0.13. A Finnish game. Also called Go-Moku. v1.0	Fred Fish Disk 116	A ram animation system with three different example animations; Kahmankas, Rocker, & F-15. Kahmankas & Rocker run on a 512K Amiga & show off overscan HAM mode. Includes a animation player program (movie), animation builder programs (idbm, plbm), & a text/graphics display program (vibm). By Eric Graham & Ken Olier
Ndr	DCDemo	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	RstNolla	V2.07 of Matt Dillon's csh like shell.S	Fred Fish Disk 117	A real neat horizontal scrolling demo that is a 2400 x 200 pixel 32 color IFF picture composed of digitized snapshots of members of the Amiga Users of Calgary, superimposed on a very wide picture of the Calgary skyline. B only. By Stephen Vermeulen & Stephen Jeans
Parse	HoDriver	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	SVTools	A util., similar to other common 'diff' programs.S. Suite provides ex. code of facilities such as FileIO Requester, XText, DoRequest, & tutorial on how to program the Amiga. Book 1.01.S	AMUC_Demo	Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on tk disk. B only. By Stephen Vermeulen
Shar	QBase	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 108	Some useful tools. S.	Fred Fish Disk 118	This is a complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. A multiplayer game of exploration, economics, war, etc., can last months. Played either by local keyboard or through modem. V1.0, shareware, & includes S code. By Chris Gray, original game by Peter Langston
SmallLib	Thai	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	AList	Dr listing prog. based on LD4 prg S	ExP_Demo	Demo version of Express Paint 1.1., used to create the scrolling demo picture in the AMUC_Demo drawer on tk disk. B only. By Stephen Vermeulen
UJencode	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	DnMaster	Disk cataloger, v1.0b, update to FF89. S.	Fred Fish Disk 119	This is a complete rewrite, from the ground up, in Draco, of Peter Langston's Empire game. A multiplayer game of exploration, economics, war, etc., can last months. Played either by local keyboard or through modem. V1.0, shareware, & includes S code. By Chris Gray, original game by Peter Langston
Fred Fish Disk #3	Thai	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Dots-Perfect	Printer Driver for an Epson MX80 printer with upgrade kit installed. S.	Fred Fish Disk 120	Displays lines whose end points are bounding around the screen, which is a double buffered HAM screen. The Y positions of the points are continuously copied into an audio waveform that is played on all four channels, & the pitch of a just intoned chord is derived from the average X position of these points. JForm. Source By Phil Burk
Dme	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	MonIDCMP	Lets you monitor the IntMessages that pass through an IDCMP window. Prints the message class, mouse coordinates, qualifier values. Great for debugging. S.	Fred Fish Disk 121	Based on original code by Leo Schwab, has credits longer than the actual demo. Runs of 512K Amiga. B only. By Hobe Oris
MicroEmacs	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	PrintPop	A util. to send common control settings to the PRT: device. S.	Fred Fish Disk 122	Demonstrates the Amiga's line drawing speed. Runs on a 512K Amiga. Includes S. By Matt Dillon
Fred Fish Disk #4	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Sectorama	Utilities to recover lost or damaged data from floppies & hard disks. v1.1, an update to FF102. V100 emulator for a Tektronix 4010/4014. (V2.6) update to FF52. S.	Fred Fish Disk 123	In Conclusion
AudioTools	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Tek	File archiver, like 'arc'. v1.24B, update to FF87	Fred Fish Disk 124	To the best of our knowledge, the materials in this library are freely distributable. This means they were either published and placed in the Public Domain by their Author, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the author's wishes, please contact us by mail.
ClickUpFront	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Zoo	A new animation.	Fred Fish Disk 125	IMPORTANT NOTICE!
HeliosMouse	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 109	A CP/M sim, simulates 8080 along with h19 emulation. S.	Fred Fish Disk 126	This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial groups only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.
FF2Ps	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Machine	Hook up your Amiga as a usernet node. S.	Fred Fish Disk 127	Any non-commercial Amiga user group wishing to duplicate this list should contact PIM Publications, P.O.Box 869, Fall River, MA 02722. PIM Publications is extremely interested in helping any non-commercial support for the Amiga.
ModulaTools	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	SimCPM	A 68000 assembler written in C. S.	Fred Fish Disk 128	
Terrain3d	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	UUpc	An optimizing C compiler for the 68000 processor. update to FF53, but not based on the code of that disk.	Fred Fish Disk 129	
Fred Fish Disk #5	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 110	A graphical monitor of cpu, bitter, & memory use. Includes two components; load device, monitors system parameters, & amyload, which is the user interface & display program. by Jeff Kelley SE	Fred Fish Disk 130	
Cmd	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	ASdk	Assigns multiple names to a given device, modified version of the original released on disk number 78. By Philip Lindsay, mod by Olaf Seibert SE	Fred Fish Disk 131	
CygnusEdDemo	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	AssgnDev	Continuously displays memory usage in a vertical bar graph Binary only. By Peter da Silva	Fred Fish Disk 132	
Gomf	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Gauge	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 133	
Journal	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	HeliosMouse	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 134	
MergelMem	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Labels	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 135	
PrinterStealer	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Mandel	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 136	
Record-Replay	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	PopLife	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 137	
Fred Fish Disk #6	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 111	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 138	
AnimPlayer	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	AnyLoad	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 139	
Cress	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	AssgnDev	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 140	
Hackbench	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Gauge	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 141	
Label	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	HeliosMouse	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 142	
LineDrawer	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Labels	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 143	
PopUpMenu	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Mandel	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 144	
Tek4695	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	PopLife	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 145	
TimeRam	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 112	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 146	
WarpText	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	BeachBlids	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 147	
Fred Fish Disk #7	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Bully	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 148	
Replaces FF57 for Copywrite problems	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	DropShadow	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 149	
QuAndPaste	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	HagenDemos	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 150	
Graphit	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Viacrom	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 151	
Juggler	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	WaveBench	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 152	
MouseReader	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 113	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 153	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	AmCron	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 154	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Dme	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 155	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 114	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 156	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 115	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 157	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 116	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 158	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 117	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 159	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 118	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 160	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 119	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 161	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 120	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 162	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 121	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 163	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 122	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 164	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 123	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 165	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 124	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 166	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 125	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 167	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 126	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 168	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 127	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 169	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 128	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 170	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 129	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 171	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 130	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 172	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 131	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 173	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 132	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 174	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 133	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 175	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 134	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 176	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 135	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 177	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 136	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 178	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 137	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 179	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 138	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 180	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 139	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 181	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 140	Pushes all open screens around (thus the name "bully"). Show more than one demo at a time By Mike Meyer S	Fred Fish Disk 182	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 141	Droptadow V2.0, use with Bryce Nesbitt's Wavebench demo. B only. By Jim MacLazav "RGB" & "Focus". RGB requires one meg. B only. By Joel Hagen	Fred Fish Disk 183	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 142	Latest version of viacrom for use in conjunction with Wavebench demo. B only. By Leo Schwab & Bryce Nesbitt	Fred Fish Disk 184	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 143	A neat screen hack, & runs on 512K machines. For more laughs, try in conjunction with Viacom or Dis (Droptadow). Includes S. By Bryce Nesbitt	Fred Fish Disk 185	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 144	Another "summouse" prog. Automatically activates a window by mouse pointer V1.1, update to disk 94. By Davide Cervone SE	Fred Fish Disk 186	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 145	Alphabetic & numeric ordered cross reference lists of defined system constants. Recommended for debugging purposes only. Use the symbolic values in prog! By Olaf Seibert	Fred Fish Disk 187	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 146	Another mandelbrot generator program, with bits & pieces of code from C. Heath & R.J. Mical. By Olaf Seibert S	Fred Fish Disk 188	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 147	A PopCLI type that plays like all over your screen. Lots of bits & pieces from Tomas Rokicki's bitlab & John Toebes' PopCLI. By Olaf Seibert S	Fred Fish Disk 189	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 148	Beach scene portrayed by sprites & sound 512K machine. By Jerrold Turner! B only.	Fred Fish Disk 190	
	Fred Fish Disk #3	WD-1002-05 hard disk controller driver. Card capable of maintaining 3 hard disks and 4 floppies, the driver is capable of only one hard disk. by Alan Kent SED	Fred Fish Disk 149			

Amaze Me

Please use this order form when subscribing to Amazing Computing™, ordering Back issues, or ordering Amiga™ Public Domain Software

Name _____
 Street _____
 City _____ St. _____ Zip _____
 Amount Enclosed _____

Please circle the appropriate item: **New Subscription** **Renewal**

Please start my subscription to Amazing Computing™ with the next available issue or renew my current subscription. I have enclosed \$24.00 for 12 issues in the U.S. (\$30.00 Canada and Mexico, \$35.00 overseas). All funds must be in U.S. Currency on a U.S. Bank

Back Issues: \$4.00 each
 (foreign orders add \$1.00 each for Postage and Handling)

Please circle your Back issue choices below:

Vol1.1	Vol1.2	Vol1.3	Vol1.4	Vol1.5	Vol1.6	Vol1.7	Vol1.8	Vol1.9	Vol2.1	Vol2.2	Vol2.3	Vol2.4	Vol2.5
Vol2.6	Vol2.7	Vol2.8	Vol2.9	Vol2.10	Vol2.11	Vol2.12	Vol3.1						

Public Domain Software:

\$6.00 each for subscribers (yes, even the new ones!)

\$7.00 each for non subscribers

Please circle your Public Domain Software choices below:

Amicus:

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A11	A12	A13
A14	A15	A16	A17	A18	A19	A20	A21	A22	A23	A24	A25	A26	

Fred Fish:

FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14
FF15	FF16	FF17	FF18	FF19	FF20	FF21	FF22	FF23	FF24	FF25	FF26	FF27	FF28
FF29	FF30	FF31	FF32	FF33	FF34	FF35	FF36	FF37	FF38	FF39	FF40	FF41	FF42
FF43	FF44	FF45	FF46	FF47	FF48	FF49	FF50	FF51	FF52	FF53	FF54	FF55	FF56
FFNA	FF58	FF59	FF60	FF61	FF62	FF63	FF64	FF65	FF66	FF67	FF68	FF69	FF70
FF71	FF72	FF73	FF74	FF75	FF76	FF77	FF78	FF79	FFNA	FF81	FF82	FF83	FF84
FF85	FF86	FF87	FFNA	FF89	FF90	FF91	FF92	FF93	FF94	FF95	FF96	FF97	FF98
FF99	FF100	FF101	FF102	FF103	FF104	FF105	FF106	FF107	FF108	FF109	FF110	FF111	FF112
FF113	FF114	FF115	FF116	FF117	FF118								

(NA denotes disks removed from the collection)

Please complete this form and mail with check or money order to:

PiM Publications, Inc.
 P.O.Box 869
 Fall River, MA 02722

Please allow 4 to 6 weeks for delivery



Multi-Prefs

Time saving utilities for EVERY Amiga owner!

- Create multiple preference settings.
- Quickly and easily restore them as needed.
- Edit, Sort, Delete and Undelete any of them.
- Fast, Fun and Easy to use, from WorkBench or CLI.
- Runs on ALL Amigas . . . 500, 1000 and 2000 (V1.2 required).
- Multi-tasking.
- Not copy protected.
- Plus . . . 3 other time saving utilities are included.

Available NOW!!!

Just send \$29.95 plus \$2.00 S&H to:

SOFT-LINK inc. P.O. Box 304, Coventry, RI 02816

Dealer inquiries invited.

Amiga is a registered trademark of Commodore-Amiga, Inc.

Index of Advertisers

A-Squared		Microillusions	CIII
Distributions Inc.	15	MicroSmiths, Inc.	87
Aegis Development	24,25	Newwave Software	30
Central Coast Software	21	NewTek	CIV
Comp-U-Save	80	Palomar Peripherals	112
Computer Mart	54	Peacock Systems, Inc.	22
Creative Solutions	67	Pioneer Computing	26
Datamax Research Corp.	33	Prolific Inc.	3
Fuller Computer Systems	32	Right Answers Group	35
Haitex Resources	17	Rittinghouse Software	
Hilton Android	63	Development Co.	28
HyperTek/		Sedona Software	49
Silicon Springs	64	Software Integration	
Inovatronics, Inc.	29	Solutions	59
Kent Engineering		Software Terminal	CII
& Design	62	SOFT-LINK inc.	112
KlineTronics	95	Speech Systems	71
Lattice, Inc.	7	Spencer Organization, Inc.	66
Lightning Publishing	37	The Memory Location	89
Megatronics, Inc.	1	The Other Guys	13
Meridian Software	70	TRU-IMAGE	3
MicroBotics, Inc.	11	Westcom Industries	93

Looking For More "Byte" For Your Dollar?



Here is the answer, the model **PPI-1000** Expansion Unit for your Amiga 1000 by **Palomar Peripherals**.

- ☐ Full auto configuration
- ☐ Meets Zorro specifications
- ☐ Amiga buss pass-thru
- ☐ Built in power supply
- ☐ Power on/off controlled by Amiga
- ☐ Real time clock with battery backup
- ☐ Fast RAM
 - 2,4,6 or 8 megabytes
 - No wait states
- ☐ Disk drives
 - Supports up to 2 drives
 - 20 or 40 megabyte hard drive(s)
 - 3 1/2" or 5 1/4" floppy drive
 - Fast DMA interface

With 20 meg hard drive and 2 meg RAM price is \$1395 plus shipping and handling (Calif. Res. add 6% sales tax). Requires Amiga DOS 1.2. For further information call **(619) 748-1202** or write to **Palomar Peripherals**.

PALOMAR PERIPHERALS

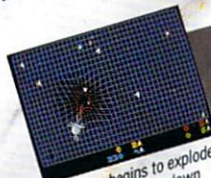
14580 High Pine
Poway, Ca 92064

EBONSTAR

EbonStar is swift action competition wrapped in an electrifying arcade style format. Exceptionally unique and challenging, EbonStar will thrill you with it's dramatic sound, realistic graphics, and options to four players and man vs man or machine. Seize the challenge of EbonStar, pick up your copy today!



The warp's fathomless depths



The warp begins to explode as opponent is shot down

ROMANTIC ENCOUNTERS

A true to life adult encounter for men or women! Permit your wildest imagination's to interact with the enticing scenarios of the DOME. At the DOME, you can set yourself up for what ever you are after... We promise you'll get what's coming to you!



One of the Dome's alluring stories



The lounge, a place of enticing rendezvous

DISCOVERY EXPANSION DISK

Learning is an exciting adventure with DISCOVERY! Now you can expand your adventure with Discovery Expansion Disks: Math 1 • MATH CONCEPTS • SPELLING 1 • SPELLING 2 • TRIVIA 1 • TRIVIA 2 • SCIENCE • GEOGRAPHY • LANGUAGE • SOCIAL STUDIES • HISTORY



Control room of space ship Discovery

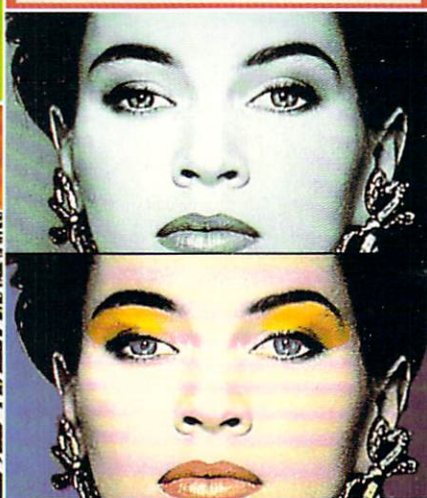
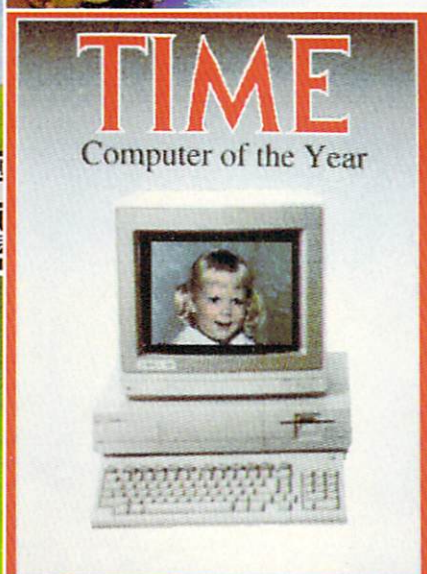
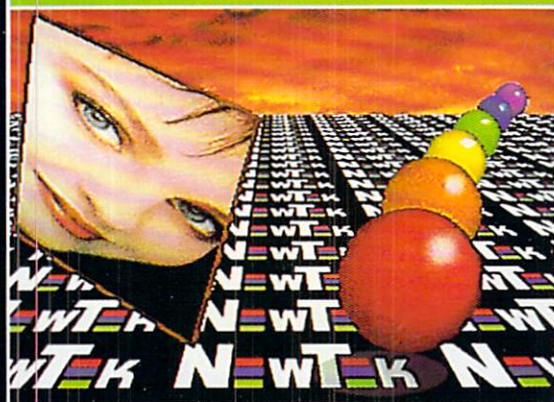


Working a solution to a math problem.

OTHER PRODUCTS FROM MICROILLUSIONS

- **PLANETARIUM™**
For the Serious Student of Astronomy
 - **LAND OF LEGENDS™**
The ultimate dungeon adventure!
 - **ONE TO ONE SERIES™**
Fire Power™, Galactic Invasion and Turbo™
 - **BLACKJACK ACADEMY**
Everything you ever wanted to know about Blackjack
 - **THE FAERY TALE ADVENTURE™**
Amiga's number one adventure game!
- All Microillusions Entertainment and Education products are developed especially for and available on Amiga, C64/128, MAC, Apple IIGS, and PC/MS DOS Formats.





ONLY DIGI-PAINT CAN DO ALL THIS

Get the maximum graphics power from your Amiga. Create stunning, lifelike computer artwork with Digi-Paint, the first full-featured 4096 color (Hold and Modify) paint program. Break the "32 color barrier" and finally realize the potential of your Amiga with Digi-Paint's advanced features:

- 4096 colors on screen simultaneously
- NewTek's exclusive enhanced HAM mode
- Dithered HAM gradient fill
- Full screen effects including double, half size, mirror reverse and more
- Full IFF and Digi-View compatibility
- Use 320x200 or HAM hi-res 320x400 resolutions
- Fat bits Magnify mode
- Rectangle, oval, line and other drawing tools
- 12 different paint modes including blending, tinting and smooth shading
- Full lasso cut and paste with automatic edge blending
- Programmed completely in assembly language for fast, smooth response

Find out why Byte Magazine called Digi-Paint "Remarkable". Available now at your local Amiga dealer or call: 1-800-843-8934.

ONLY \$59.95

NewTek
INCORPORATED